

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"  
УДК 004.415.25

«До захисту допущено»  
Завідувач кафедри  
О.В. Коваль  
(підпис) (ініціали, прізвище)  
“ ” 2019р.

Магістерська дисертація

зі спеціальності - 121 Інженерія програмного забезпечення  
за спеціалізацією - Інженерія програмного забезпечення розподілених систем  
на тему: «Галузевий реєстр електронних інформаційних ресурсів»

Виконав (-ла): студент (-ка) 6 курсу, групи ТВ-81мп  
Іванюк Андрій Олегович  
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент, Кузьменко І. М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент  
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць інших  
авторів без відповідних посилань.

Студент  
(підпис)

Київ – 2019

**Національний технічний університет України  
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Програмне забезпечення автоматизованих систем

за спеціалізацією - Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Коваль О.В.  
(прізвище, ініціали) \_\_\_\_\_ (підпис)  
« \_\_\_\_ » \_\_\_\_\_ 2019р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Іванюку Андрію Олеговичу

(прізвище, ім'я, по батькові)

1. Тема дисертації “Галузевий реєстр електронних інформаційних ресурсів”

Науковий керівник Кузьменко Ігор Миколайович к.т.н., доцент.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету № \_\_\_\_ від “ \_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_ року

2. Строк подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження процес переповнення даних в системах управління електронними інформаційними ресурсами

4. Предмет дослідження атрибути електронних інформаційних ресурсів

5. Перелік питань, які потрібно розробити \_\_\_\_\_

1) проаналізувати особливості роботи галузевих реєстрів електронних інформаційних ресурсів;

2) забезпечити веб-інтерфейс для доступу та керування галузевим реєстром;

3) розробити гнучкий алгоритм регулювання доступу до електронних інформаційних ресурсів у реєстрі;

4) забезпечити обширний набір пошуку електронних інформаційних

ресурсів. \_\_\_\_\_

6. Орієнтований перелік ілюстративного матеріалу \_\_\_\_\_

1) Діаграма прецедентів

2) Архітектура бази даних

3) Функції програмного забезпечення

4) Етапи роботи з програмою

5) Веб-інтерфейс

7. Орієнтований перелік публікацій \_\_\_\_\_

1) Іванюк А.О., Кузьменко І.М. “Веб-система для управління реєстром електронних інформаційних ресурсів” \_\_\_\_\_

8. Дата видачі завдання « 2 » вересня \_\_\_\_\_ 2019 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Терміни виконання етапів магістерської дисертації	Примітка
	Отримання завдання	02.09.19р.	
	Збір інформації	02.09.19р. – 09.09.19р.	
	Аналіз вимог завдання, вибір методів і засобів розв’язання поставленої задачі	09.09.19р. – 16.09.19р.	
	Підготовка публікацій	17.09.19р., 25.09.19р.	
	Підготовка доповідей на конференції	25.09.19р., 27.09.19р.	
	Підготовка дисертації	30.09.19р. – 10.10.19р.	
	Розробка програмного продукту	10.10.19р. – 24.10.19р.	
	Захист програмного продукту	25.10.19р.	
	Передзахист	22.11.19р.	
	Захист	18.12.19р.	

Студент

\_\_\_\_\_  
( підпис )

**Іванюк А.О.**

\_\_\_\_\_  
(прізвище та ініціали)

Науковий керівник

\_\_\_\_\_  
( підпис )

**Кузьменко І.М.**

\_\_\_\_\_  
(прізвище та ініціали)

# РЕФЕРАТ

## Структура й обсяг дипломної роботи

Магістерська дисертація складається зі вступу, 6 розділів, висновку, переліку посилань з 26 найменувань, 1 додатку і містить 22 рисунків, 24 таблиці. Повний обсяг магістерської дисертації складає 83 сторінок, з яких перелік посилань займає 3 сторінок, додатки – 10 сторінок.

Актуальність теми. Навіть після двох десятиліть цифрової присутності та розповсюдження електронних ресурсів з точки зору створення, придбання та використання, управління ними залишається громіздким процесом. Процес управління електронними ресурсами часто перенавантажує людей. Життєвий цикл електронних ресурсів, особливо в академічних установах, починається з відкриття та ідентифікації ресурсу та переходить до доступу на засадах випробувального терміну, вибору конкретного ресурсу з гами інших ресурсів з подальшим придбанням, а потім охоплює доступ до користувачів, а потім подальше вивчення використання ресурсу для подальшої підтримки та актуалізації цих ресурсів. Між цими процесами стоїть громіздка робота про перегляд ліцензійних угод та правильність ведення записів. Усі ці процеси займають багато часу і вимагають великої праці. Ці невблаганні масиви роботи змусили шукати системи, які дозволять заощадити свій час та енергію, а також забезпечать ефективне управління електронними ресурсами.

**Мета створення** полягає в побудові та розвитку інформаційного простору, для використання в прикладній галузі, а також забезпеченні реєстрації, обліку, збереженні та захисту існуючих, придбаних або переданих на законних підставах відомостей про електронні інформаційні ресурси (EIP) з відкритою інформацією.

Також дуже важливим було забезпечення користувачам (споживачам) оперативного доступу до відомостей про EIP з відкритою інформацією, які занесені до Реєстру.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати існуючі рішення, які працюють з галузевими реєстрами;

- проаналізувати існуючі підходи класифікаційних моделей, які в перспективі можна інтегрувати в розроблювану систему для класифікацій електронних ресурсів;
- розробити пошукову модель EIP;
- розробити модель, яка буде забезпечувати регульований доступ до ресурсів, в залежності від рівня їх секретності.

**Об'єктом дослідження** є процес переповнення даних в системах управління електронними інформаційними ресурсами, що породжує труднощі при пошуку необхідної інформації.

**Предметом дослідження** атрибути електронних інформаційних ресурсів, на основі яких будуються класифікаційні та пошукові алгоритми.

**Методи дослідження.** Розв'язання поставлених задач виконувались з використанням наступних методів:

- метод порівняльного аналізу розроблюваної системи з існуючими рішеннями;
- метод симуляції робочого навантаження на систему, і дослідження її поведінки.

**Новизна одержаних результатів.** Найбільш суттєвими результатами магістерської дисертації є удосконалення механізмів для класифікації та пошуку електронних інформаційних ресурсів.

**Практичне значення одержаних результатів** роботи полягає в розробленій прикладній програмі, яка спрямована на ефективний облік галузевими реєстрами, та розроблених алгоритмах пошуку та класифікацій електронних інформаційних ресурсів.

**Ключові слова.** ГАЛУЗЕВИЙ РЕЄСТР, ЕЛЕКТРОННИЙ ІНФОРМАЦІЙНИЙ РЕСУРС.

# ABSTRACT

## **Structure and volume of the dissertation**

Master's dissertation consists of introduction, 6 sections, conclusion, list of sources of 26 items, 1 appendix and has 22 images, 24 tables. Whole dissertation volume is 83 pages, out of which list of sources takes 3 pages, appendixes - 10 pages.

## **Significance of the topic.**

Even after two decades of digital presence and the proliferation of electronic resources in terms of creation, acquisition and use, managing them remains a cumbersome process. The process of managing electronic resources often overloads people. The lifecycle of electronic resources, especially in academic institutions, begins with the discovery and identification of the resource and proceeds to trial access, selecting a specific resource from a range of other resources with subsequent acquisition, and then encompassing user access, and then further exploring the use of the resource for further supporting and updating these resources. These processes include the cumbersome work of reviewing license agreements and keeping records properly. All these processes are time consuming and require a lot of work. These inexorable arrays of work have forced them to look for systems that will save their time and energy, as well as ensure efficient management of electronic resources.

**The purpose of the creation** is to build and develop an information space for use in the application field, as well as to secure the registration, accounting, preservation and protection of existing, acquired or legally disclosed information on electronic information resources (EIR) with open information.

It was also very important to provide users (consumers) with prompt access to information on the EIA with open information entered in the Register.

To achieve this task, the following **research objectives** were formulated, which defined the logic of the study and its structure:

- analyze existing solutions that work with industry registries;
- to analyze the existing approaches of classification models that can be integrated into the developed system in the future for the classification of the desired electronic resources;

- develop an EIR search model;
- develop a model that will provide regulated access to resources, depending on their level of secrecy.

The **object** of the exploration is the process of data overflow in electronic information resource management systems, which creates difficulties in finding the necessary information.

The **subject** of the exploration is attributes of electronic information resources, which are used to build classification and search algorithms .

**Exploration methods.** Solution of defined goals was made with usage of next methods:

- method of comparative analysis of already existing solutions;
- method of simulating the workload on a system and investigating its behaviour.

**Novelty of the results.** The most valuable scientific results of masters dissertation is improving the mechanisms for classification and search of EIR.

**Practical value** of the obtained results of the work lies in the developed application program, which is aimed at the efficient accounting of industrial registers, as well as the developed search algorithms and classification of electronic information resources.

**Key words.** INDUSTRIAL REGISTRY, ELECTRONIC INFORMATION RESOURCE

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ ..	10
ВСТУП .....	11
1. ЗАВДАННЯ РОЗРОБКИ СИСТЕМИ ДЛЯ КЕРУВАННЯ ГАЛУЗЕВИМ РЕЄСТРОМ ТА ОСНОВНА МЕТА РОБОТИ .....	12
2. АНАЛІЗ РЕЄСТРУ ІНФОРМАЦІЙНИХ РЕСУРСІВ .....	13
2.1. Визначення інформаційного електронного ресурсу .....	13
2.2. Визначення галузевого реєстру та приклади існуючих реєстрів .....	14
2.3 Структура і функціонування типового реєстру .....	15
2.4 Висновки до розділу 2 .....	17
3. ТЕХНОЛОГІЇ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ.....	19
3.1. Система керування базами даних Microsoft SQL Server.....	19
3.2. Мова програмування C# і платформа .NET Core .....	22
3.3. Використання фреймворку ASP.NET Core .....	23
3.4. Технологія Entity Framework Core .....	24
3.5. Технологія контейнерів Linux та Docker .....	27
3.6. Трирівнева архітектура .....	30
3.7 Сервіс балансування NGINX .....	32
3.8. Висновки до розділу 3 .....	33
4. ДЕКОМПОЗИЦІЯ СИСТЕМИ ТА ОСОБЛИВОСТІ АРХІТЕКТУРИ .....	35
4.1. Опис програмної системи на прикладі діаграми прецедентів.....	35
4.2. Взаємодія програмної системи із зовнішніми сервісами.....	36
4.3. Відокремлення інфраструктурних одиниць архітектури.....	37
4.4. Відокремлення основних модулів в програмній реалізації .....	38
4.5. Опис схеми бази даних.....	40
4.6. Висновки до розділу 4 .....	43
5. РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ .....	45
5.1. Вимоги для функціонування системи.....	45
5.2 Інструкція для роботи з системою .....	45



5.3. Висновки до розділу 5 .....	49
6.1 Опис ідеї проекту .....	51
6.2 Технологічний аудит ідеї продукту .....	54
6.3 Аналіз ринкових можливостей запуску стартап-проекту .....	55
6.4 Аналіз ринкової стратегії проекту .....	63
6.5 Розробка маркетингової програми стартап-проекту .....	65
6.6 Висновки до розділу 6 .....	70
ВИСНОВКИ .....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	72
ДОДАТОК А .....	75

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ГР – галузевий реєстр

ERMS (electronic resources management system) – система управління електронними ресурсами

ОТР (one-time password) – одноразовий пароль

ЕІР – електронний інформаційний ресурс

СІР – система інформаційних ресурсів

ПЗ – програмне забезпечення

## ВСТУП

Галузевий реєстр електронних інформаційних ресурсів – це інформаційна система, метою якої є реєстрації, обліку, акумулювання, оброблення і зберігання даних про структуру, зміст, розміщення, рівень доступу до електронних інформаційних ресурсів та виконання основних вимог галузевого міністерства.

Дивлячись на створення в країні основ інформаційного суспільства, вагомого значення набуває розробка систем управління галужевими реєстрами, ціллю яких є створення системи, здатної покращити якість експлуатації електронних ресурсів та суттєво зменшити витрати на їх утримання та операційні витрати за рахунок використання метаінформації щодо інформаційних масивів, сховищ, реєстрів та інших видів інформаційних ресурсів, а зокрема тих, що утворюються за державні кошти в різних національних установах.

Варто зазначити, що при розробці такої системи з'являються проблеми, спричинені відсутністю одного і визначеного способу зберігання та обробки електронних ресурсів, і відсутності уніфікованого інтерфейсу доступу до них. Крім того, важливим є те, що не існує стандартизованого механізму упорядкування та кодування даних, а також достатньої кількості спеціалістів з достатньою кваліфікацією, які вміли б працювати з електронними ресурсами та даними.

Тому ціллю роботи є розробка системи, яка могла б вирішувати проблеми, пов'язані з уніфікованим доступом до ресурсів, інформаційною безпекою, а також могла б якісно керувати галузевим реєстром та інтегрувати електронні ресурси міністерства, використовуючи атрибути ресурсів та, як наслідок, покращити ефективність їхньої експлуатації.

# 1. ЗАВДАННЯ РОЗРОБКИ СИСТЕМИ ДЛЯ КЕРУВАННЯ ГАЛУЗЕВИМ РЕЄСТРОМ ТА ОСНОВНА МЕТА РОБОТИ

Метою роботи є створення системи для управління галузевим реєстром, яка зможе вирішити задачі, пов'язані з єдиним уніфікованим способом доступу до електронних документів, безпекою даних, і могла б ефективно керувати реєстром та інтегрувати електронні ресурси галузевого міністерства, використавши схему метаданих ресурсів і, як результат, покращити ефективність їхньої експлуатації. Програмна система повинна містити наступні функції:

- занесення електронного ресурсу в систему;
- оновлення та перегляд змісту електронних ресурсів галузевого реєстру;
- завантаження вмісту ресурсу;
- розширений пошук ресурсів по їх атрибутах;
- класифікація електронних ресурсів;
- реєстрація користувача в системі;
- двох-факторна автентифікація та авторизація користувача;
- підтвердження запиту користувача на реєстрацію;
- можливість горизонтального масштабування системи, в залежності від

робочого навантаження.

Для виконання поставленої задачі необхідно дослідити інструментарій і стек технологій для реалізації, проаналізувати існуючі галузеві реєстри інформаційних ресурсів, дослідити можливі варіанти та спосіб, в який будуть зберігатися дані, розробити схему метаданих галузевого реєстру, та розробити систему для доступу до даних галузевого міністерства, забезпечити механізм для регулювання рівня доступу до ресурсів, їх реєстрацію і видалення, а також забезпечити модульну структуру системи, що дасть можливість з легкістю підтримувати функціонал існуючої програми, а також оновлювати і перебудовувати вже встановлену архітектуру.

В результаті програма буде використовуватися для ефективного керування галузевим реєстром враховуючи дозволи користувача та рівень секретності ресурсів.

## **2. АНАЛІЗ РЕЄСТРУ ІНФОРМАЦІНИХ РЕСУРСІВ**

Роздивимось визначення поняття галузевого реєстру та реєстру. Також проаналізуємо існуючі реалізації таких систем.

### **2.1. Визначення інформаційного електронного ресурсу**

Реєстр електронних ресурсів – це деяка система, призначена для зберігання, класифікації та організації даних в оптимізований спосіб, з метою полегшення їх зберігання та подальше опрацювання [5].

Інформаційні ресурси можна класифікувати як:

- окремі документи, файли і масиви файлів;
- продукти розумової діяльності людини;
- бази даних та інші сховища, в незалежності від способу зберігання даних.

Любі інформаційні ресурси, для відновлення змісту яких потрібен електронний пристрій, можна назвати електронним інформаційним ресурсом.

“Дедалі ширше використання електронних документів вимагає розробки практичної системи їх класифікації - такої, що забезпечуватиме адекватну і зрозумілу передачу змісту документів з метою їх ідентифікації, упорядкованого зберігання та швидкого знаходження в електронних ресурсах, передбачатиме класифікацію і пошук за різними критеріями (відповідно до специфіки електронних пошукових систем), дозволить вільно оперувати електронними документами незалежно від зміни їх типів, видів тощо” [1].

За станом електронні ресурси можна класифікувати на наступні види:

- першоджерела;
- оцифровані варіанти звичайних видань;
- оцифровані аналоги звичайних видань;
- копії та інші відтворення першоджерел та оригіналів

Можна побачити, що у цих трактуваннях схожим буде підкреслення важливості

наявності споживання ресурсу людьми, як деякої інформації. Проте варто зазначити, що О. В. Соснін вважає, що ресурс було створено базуючись на даних, яка були раніше зафіксовані та оцінені: «Для того, щоб стати інформаційним ресурсом, потоки інформації повинні мати деякі специфічні якості, завдяки яким вони стають соціально значущими, технологічно придатними, тобто такими, що мають цінність для практичного застосування. Такою основною якістю є системна організованість (організація) інформаційних потоків та їх окремих елементів» [2].

Проводячи аналіз між різними способами дати поняття ресурсу можна сказати, що основною характерною рисою, яка відрізняє їх від документу «Про Національну програму інформатизації» [3], є заміна понять “файл” або “документ” на інші. Професор О. В. Соснін зазначає: «Інформаційний ресурс це інформація, яка створена чи виявлена, в установленому порядку зареєстрована, оцінена, яка має визначені закони деградації і оновлення, набуває особливих властивостей, які становлять її сутність і роблять її інформаційним продуктом, з одного боку, і предметом споживання – з іншого».

## **2.2. Визначення галузевого реєстру та приклади існуючих реєстрів**

Щоб дати визначення поняттю галузевого реєстру, важливим є розуміння того факту, що кожна галузь характеризується унікальною структурою, яка може динамічно змінюватись в часі, передбачати різні рівні доступу до ресурсів і каталогів, виходячи з специфічних проблем предметної області.

“Реєстр – це інформаційно-телекомунікаційна система, призначена для реєстрації, обліку, накопичення, оброблення і зберігання відомостей про склад, зміст, розміщення, умови доступу до електронних інформаційних ресурсів та задоволення потреб юридичних і фізичних осіб в інформаційних послугах” [5].

Галузевим можна назвати будь-який реєстр, який може відобразити структуру певної галузі, здатний забезпечити всі інформаційні потреби в достатньому обсязі і

може надати необхідний рівень безпеки для даної інформації.

Яскравими прикладами існуючих галузевих реєстрів є:

- Online Computer Library Center (OCLC) – некомерційна організація, яка надає бібліотечні послуги та займається науково-дослідною роботою. Одною з основних спеціалізацій організації є збільшення доступу до даних, і зменшення витрат для керування даними.
- GSA Register – компанія, яка забезпечує цілісність, управління та зберігання даних про нерухомість.
- Federal Library and Information Network (FEDLINK) – організація федеральних агенств, працюючих разом, щоб досягти оптимального використання ресурсів і можливостей використання бібліотек та інформаційних центрів, пропонуючи сервіси, координування і розподіл доступних ресурсів.

## 2.3 Структура і функціонування типового реєстру

Реєстр EIP з відкритою інформацією інтегрується з Національним реєстром електронних інформаційних ресурсів із забезпеченням доступу до них через мережу Інтернет.

Основними складовими галузевого реєстру є робочий, страховий, еталонний та інформаційні фонди.

Як відомо, еталонні інформаційні ресурси являють собою системи, призначені для зберігання та обліку відомості та інформації про розміщення та умови доступу до EIP.

Варто також зазначити, що подібною системою, яка опрацьовує дані про структуру, знаходження та рівень доступу до ресурсу з відкритими даними є робочий фонд.

Страховий фонд EIP – архівні копії еталонного фонду EIP з відкритою інформацією, ціллю якого є відновлення стану системи у разі збою роботи системи.

Інформаційний фонд EIP – програма, яка, подібно до страхового фонду, містить

копії еталонного фонду.

У реєстрі реалізуються функції обліково-реєстраційні, класифікаційні, інформаційно-пошукові та адміністративні.

До обліково-реєстраційних функцій відносяться:

- занесення інформації про ЕІР;
- редагування інформації про ЕІР;
- видалення інформації про ЕІР.

До класифікаційних функцій відносяться:

- перевірка тематичної належності ЕІР;
- створення інформативних ключових витримок;
- автоматизована класифікація даних.

До інформаційно-пошукових функцій відносяться:

- пошук ЕІР за каталогами (тематичні рубрики, регіони, власники);
- пошук ЕІР за їх атрибутами;

До адміністративних функцій відносяться:

- керування масштабованою базою даних;
- керування змістом інформаційних ресурсів
- актуалізація даних Реєстру;
- резервне копіювання відомостей про ЕІР.

Для ведення Реєстру електронні інформаційні ресурси з відкритою інформацією класифікуються за наступними характеристиками:

- рівень доступності ресурсу;
- міра послідовності і логічності змісту ресурсу;
- автор ресурсу.

Під час проведення перевірки фактичної наявності і складу ЕІР повинні вирішуватися наступні завдання:

- виявлення фактичної наявності ЕІР: класифікація ЕІР;
- визначення вартості ЕІР;
- визначення категорії ЕІР за порядком доступу до інформації (відкрита



інформація) та віднесення до реєстру з відкритою інформацією;

- визначення власника EIP;
- визначення користувачів, яким надається доступ до EIP;
- встановлення регламентації способів зберігання EIP (відповідно до типу реєстру);
- аналіз поточного етапу EIP;
- документування результаті в інвентаризації EIP.

## 2.4 Висновки до розділу 2

Результати проведеного аналізу відображені в таблиці (таблиця 2.1).

Таблиця 2.1 – порівняльний аналіз існуючих рішень

	OCLC	FED LINK	GSA REGISTER	Розроблена система
Безплатна експлуатація	+	-	+	+
Простота знаходження необхідних ресурсів	+	+	+/-	+
Сповіщення користувачів про зміни в системі через EMAIL та SMS	+	+	-	+
Можливість здійснювати класифікацію електронних ресурсів	+	+	+	+
Можливість регулювання рівня доступу до окремих ресурсів і каталогів для користувачів	-	+	-	+
Незалежність від галузі експлуатації	-	-	+	+

Можна побачити, що ніяка система крім GSA Register не може використовуватись незалежно від контексту конкретної галузі. Проте і в ній є деякі недоліки, як обмеженість можливостей розширеного пошуку та ідентифікацію

необхідного ресурсу. Також система GSA Register, подібно до OCLC, не дають змогу регулювати рівень доступу до ресурсів.

Розроблена система управління галузевими реєстрами реалізує всі функції, наведені вище (таблиця 2.1), що дає їй перевагу перед існуючими рішеннями.

## 3. ТЕХНОЛОГІЇ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

### 3.1. Система керування базами даних Microsoft SQL Server

MS SQL Server – є одною з найбільш популярних систем керування базами даних в світі. Дана система підходить для різних проектів: від невеликих застосунків до великих високонавантажених систем.

SQL Server був створений компанією Microsoft. Перша версія з'явилась ще в 1987 році. В даній роботі була використана 16 версія серверу.

SQL Server довгий час був системою керування базами даних тільки для Windows, однак з версії 16 став доступним і на інших операційних системах.

MS SQL характеризується наступними особливостями:

- продуктивність – SQL Server відзначається дуже швидким доступом до даних;
- надійність і безпечність – дана технологія надає можливість шифрування даних;
- простота – з даною СКБД відносно легко працювати і вести адміністрування.

Центральним аспектом в MS SQL Server, як і в іншій СКБД є база даних. Вони визначає сховище даних, організованих в певний спосіб. Не є рідкістю те, що база даних представляє файл на диску, хоча така асоціація не є обов'язковою. Для зберігання і адміністрування баз даних застосовуються системи керування – СКБД. MS SQL як раз і є одною з таких систем керування.

Для організації баз даних MS SQL використовує реляційну модель. Ця модель баз даних була розроблена ще в 1970 році Едгаром Кодом. А на сьогоднішній день вона є фактично стандартом для організації баз даних.

Реляційна модель передбачає зберігання даних у вигляді таблиць, кожна з яких складається із рядків і стовпчиків. Кожний рядок зберігає окремий об'єкт, а в

стовпцях розміщуються атрибути цього об'єкта.

Для ідентифікації кожного рядка в рамках таблиці застосовується первинний ключ (primary key). В якості первинного ключа може виступати один або декілька стовпців. Використовуючи цей ключ, можна посилатися на певний рядок в таблиці. Відповідно два рядки не можуть бути представлені у вигляді відношення.

Для взаємодії з базою даних застосовується мова SQL (Structured Query Language). Клієнт (наприклад зовнішня програма) відправляє запити на мові SQL засобами спеціального API. СКБД належним чином інтерпритує і виконує запит, а потім посилає клієнту результат виконання.

З самого початку мова SQL була розроблена в компанії IBM для системи бази даних, яка називалась System/R. При цьому сама мова називалась SEQUEL (Structured English Query Language). Хоча в кінцевому результаті, ні база даних, ні сама мова не були офіційно опубліковані.

В 1979 році компанія Relational Software Inc. розробила першу систему керування баз даних, яка називалася Oracle і використовувала мову SQL. В зв'язку з успіхом даного продукту компанію було перейменовано в ORACLE.

Внаслідок почали з'являтися інші системи баз даних які використовували SQL. В 1989 році Американський Національний інститут Стандартів (ANSI) визначив мову і опублікував перший стандарт для неї. Після цього стандарт періодично поновлювався і доповнювався. Останнє оновлення відбулось в 2011 році. Не дивлячись на наявність стандарту нерідко виробники СКБД використовують власні реалізації мови SQL, які трохи відрізняють один від одного.

Виділяють два різновиди мови SQL: PL-SQL та T-SQL. PL-SQL використовується в таких СКБД як ORACLE і MYSQL. T-SQL (Transact SQL) застосовується в SQL Server.

В залежності від задач, які виконує команда T-SQL, вона може належати одному з наступних видів:

- DDL (Data Definition Language) – мова визначення даних. До цього типу відносяться різні команди, які створюють базу даних, таблиці індекси, збережені процедури. До цього типу можна віднести наступні команди: Create, Alter, Drop,

Truncate.

- DML (Data Manipulation Language) – мова маніпулювання даними. До цього типу відносяться команди для отримання даних, їх оновлення, додавання, видалення – в цілому, всі команди, з допомогою яких можна керувати даними. До цього типу можна віднести наступні команди: Select, Update, Insert, Delete.

- DCL (Data Control Language) – мова керуванням доступу до даних. До цього типу відносять команди, які керують правами доступу до даних. Прикладом можуть бути команди Grant та Revoke.



Рисунок 3.1 – Основні компоненти MSSQL[12]

На ілюстрації вище (рисунок 3.1) розглянуті основні компоненти архітектури MS SQL Server.

## 3.2. Мова програмування C# і платформа .NET Core

Мова C# - проста, сучасна мова об'єктно-орієнтованого програмування з безпечною типізацією. Вона відноситься до широкого кола C-подібних мовних сімейств. Мова також підтримує компонентно-орієнтований підхід програмування. Розробка сучасних застосунків все більше вимагає створення компонентів в формі автономних і само-описуючих пакетів, які реалізують окремі функціональні можливості. Важлива особливість таких компонентів полягає в тому, що це є моделлю програмування на основі властивостей, методів і подій. Кожний компонент має атрибути, які являють собою декларативну інформацію про компонент, а також вбудовані елементи документації. C# надає мовні конструкції, які безпосередньо підтримують таку концепцію роботи. Завдячуючи цьому, C# добре підходить для створення і застосування програмних компонентів.

Наступні функції мови C#, які забезпечують надійність і стабільність застосунків:

- збір мусору – автоматичне звільнення пам'яті, яка займається знищеними та невикористовуваними об'єктами;
- обробка виключних ситуацій – представлення структурованого і розширеного способу виявляти і опрацьовувати помилки;
- строга типізація – особливість, яка не дозволяє оперувати неініціалізованими змінними, виходити за межі індексних масивів або виконувати неконтрольоване приведення типів.

В C# існує єдина система типів. Всі типи в мові, включаючи примітиви, такі як `int` чи `double`, успадковують від базового типу `object`. Таким чином, всі типи використовують спільний набір операцій і значення будь-якого типу можна зберігати, передавати і опрацьовувати в схожий спосіб. Крім того, C# підтримує користувацькі типи-посилання і типи значень, дозволяючи динамічно виділяти пам'ять для об'єктів і зберігати спрощені структури даних в стеку.

Щоб забезпечити сумісність програм і бібліотек C# при подальшому розвитку, при розробці C# багато уваги було приділено керуванню версій. Багато мов

програмування незважають на цей аспект, і в результаті програми на цих мовах ламаються при появі нової версії бібліотеки.

### 3.3. Використання фреймворку ASP.NET Core

ASP NET Core – технологія від компанії Microsoft, призначенням якої є створення веб-застосунків різного виду, починаючи від маленьких веб-сайтів, закінчуючи великими веб-порталами і веб-сервісами.

З одного боку, ASP NET Core представляє собою розвинуту платформу ASP.NET, а з іншого якісну зміну всієї платформи.

Розробка над платформою почалась ще в 2014 році. Тоді платформа умовно називалась ASP.NET vNext. В червні 2016 року вийшов перший реліз платформи. А в серпні 2019 року вийшла версія ASP.NET Core 3.0, яка і була використана під час реалізації системи управління галузевими реєстрами.

ASP.NET Core є повністю відкритим opensource-фреймворком. Дана технологія може працювати поверх крос-платформного середовища .NET Core, яка може розгортатись на основних популярних операційних системах: Windows, Mac OS та Linux. Таким чином, з допомогою ASP.NET Core можна створювати крос-платформні програми. Проте варто зазначити, що Windows, в якості середовища розробки і розгортання застосунків до цих пір є лідером. Але основна перевага даної технології в тому, що тепер немає привязки до однієї операційної системи. Це означає, що можна запускати веб-застосунки на операційних системах Linux та Mac OS. А для розгортання програм можна використовувати IIS сервер, або крос-платформний веб-сервер Kestrel.

Завдяки модульності фреймворку, всі необхідні компоненти веб-застосунку можуть загрузатися як окремі модулі через пакетний менеджер Nuget.

ASP NET Core включає в себе фреймворк MVC, який об'єднує в собі функціональність патернів MVC, Web API та Web Pages. В минулих версіях платформи дані технології реалізовувались окремо, і тому містили багато

функціоналу який дублювався. Зараз вони об'єднані в одну програмну модель ASP.NET Core MVC.

ASP NET Core характеризується можливістю розширюваності. Фреймворк побудований з цілого набору відносно незалежних компонентів, і дає можливість використовувати їх вбудовану реалізацію, або доповнити їхню функціональність з допомогою механізму успадковування, або ж взагалі створити і застосовувати свої компоненти зі специфічним функціоналом.

Фреймворк ASP NET Core був вибраний з огляду на наступні міркування:

- продуктивність роботи – з доданням нових бібліотек та оптимізацією інструментів фреймворку, з'явилась можливість писати ефективний код який легко підтримувати. Більше того, завдячуючи появі веб-серверу Kestrel, даний фреймворк тепер є одним з найшвидших на ринку;
- крос-платформна реалізація – на відміну від .NET, який розроблювався для того щоб запускатися на Windows, NET Core може запускати також і на інший операційних системах;
- підтримка контейнеризації – можливість запускати додаток в контейнерах Docker, та системах оркестрації контейнерами Kubernetes, додає ще більше гнучкості при розробці застосунків;
- модульний і відкритий фреймворк – гнучкість платформи з відкритим вихідним кодом дає переваги для розробці будь-яких систем.

### **3.4. Технологія Entity Framework Core**

Технологія Entity Framework Core являє собою об'єктно-орієнтовану, компактну і розширювану технологію від компанії Microsoft для доступу до даних. EF Core є ORM-інструментом (object-relational mapping – відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати з базами даних, представляючи при цьому високий рівень абстракції: технологія дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з різними технологіями збереження даних. Якщо



на фізичному рівні програма оперує таблицями, індексами, унікальними та зовнішніми ключами, то на концептуальному рівні, який пропонується EF Core, можна оперувати об'єктами.

EF Core підтримує багато різних систем керування базами даних. Таким чином можна працювати з будь-якою базою даних, якщо для неї є необхідний провайдер. По замовчуванню на даний момент Microsoft надає ряд вбудованих провайдерів: для роботи з MS SQL Server, для SQLite, для PostgreSQL. Також є провайдери від сторонніх постачальників, таких як MySQL.

Також варто зазначити, що EF Core надає універсальний API для роботи з даними. І якщо необхідно буде, для прикладу, змінити цільову СКБД, то основні зміни в проекті будуть стосуватися перш за все конфігурації і налаштувань підключення до відповідних провайдерів. А код, який безпосередньо працює з даними не зміниться.

Entity Framework Core успадкував багато від своїх попередників, зокрема, Entity Framework 6. В той же час потрібно розуміти, що EF Core – це не нова версія по відношенню до EF6, а зовсім інша технологія, хоча в цілому принципи роботи у них співпадають. Тому в рамках EF Core використовується своя система версій. При реалізації даної роботи була використана версія 3.0.

Як технологія доступу до даних, Entity Framework Core може використовуватися на різних платформах стеку .NET. Це можуть бути стандартні платформи, типу Windows Forms, консольні застосунки, UWP, WPF і ASP.NET Core. При цьому крос-платформна природа дозволяє задіяти її не тільки на операційній системі Windows, а і на Linux і Mac OS.

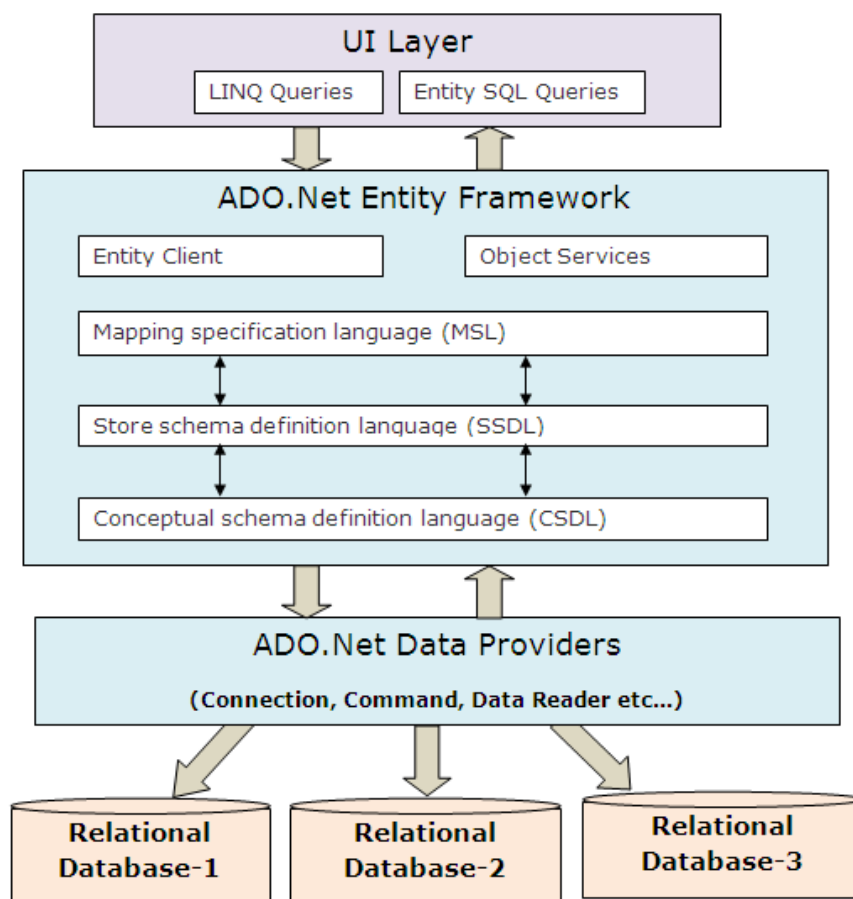


Рисунок 3.2 — Архітектура доступу до даних

На ілюстрації вище (рисунок 3.2) зображена схема роботи технології EF Core. Центральною концепцією фреймворку є поняття сутності або entity. Сутність визначає набір даних, які зв'язані з певним об'єктом. Тому дана технологія розраховує на роботу з об'єктами і колекціями, а не з таблицями.

Будь-яка сутність, як ілюбий об'єкт з реального світу, має деякий ряд властивостей. Наприклад, якщо сутність описує людину, то можна виділити такі властивості як ім'я, прізвище, вік, ріст. Властивості необов'язково повинні представляти такі типи даних як int чи string, але можуть також визначати і більш складні типи даних. Кожна сутність(entity) може містити одну або декілька властивостей, які унікально ідентифікують її.

Відмінною рисою Entity Framework Core, як технології ORM, є використання запитів LINQ для вибірки даних з бази даних. З допомогою LINQ можна створювати різні запити для вибірки об'єктів, в тому числі зв'язаних асоціативними зв'язками. А Entity Framework Core при виконанні запиту транслює LINQ в вирази, зрозумілі для

конкретної СУБД.

### 3.5. Технологія контейнерів Linux та Docker

Технологія Docker — технологія віртуалізації, яка працює з допомогою контейнерів Linux.

Для ізоляції процесів на одній фізичній машині, запуску застосунків, призначених для різних платформ, можна використовувати віртуальні машини. Віртуальні машини ділять між собою наступні фізичні ресурси хоста:

- пам'ять;
- процесор;
- дисковий простір;
- мережеві інтерфейси.

На кожній віртуальній машині встановлюється потрібна операційна система і запускається застосунок. Недоліком такого підходу є те, що значна частина ресурсів машини витрачається не на “корисне навантаження” (робота програм), а на роботу декількох операційних систем.

Альтернативним підходом до ізоляції застосунків є контейнери. Саме поняття контейнера не нове і давно відоме в Linux. Ідея в тому, що в рамках однієї операційної системи виділити ізольовану область і запускати в ній програми. В цьому випадку йде мова про віртуалізацію на рівні ОС. На ілюстрації нижче (рисунок 3.3) зображені відмінності традиційних віртуальних машин від контейнерів.

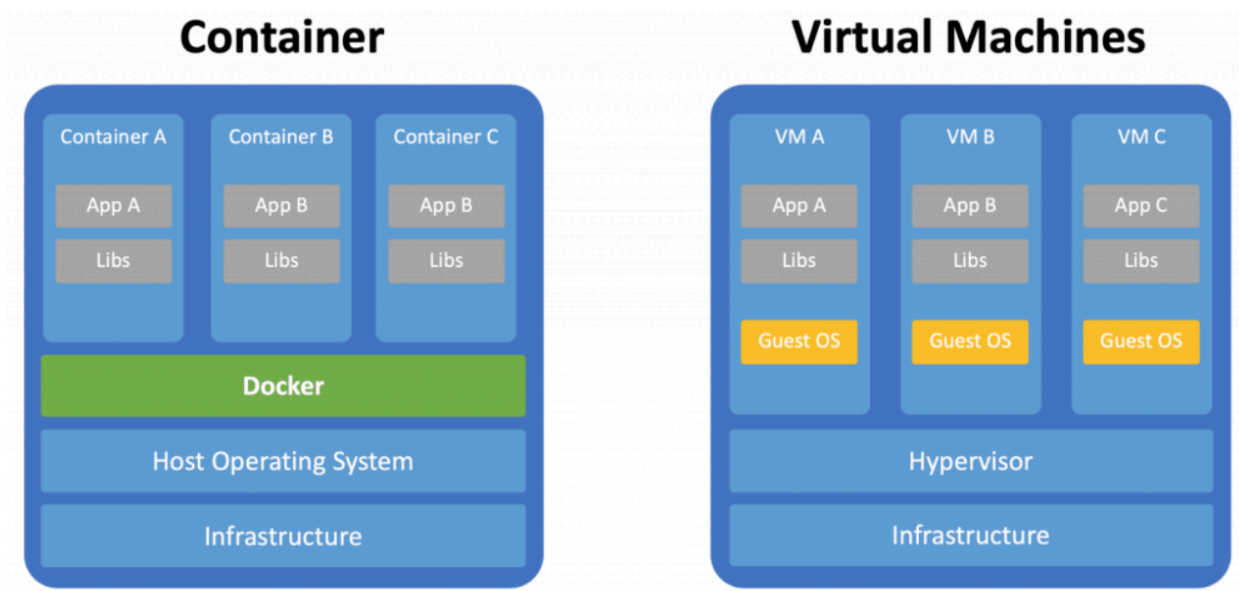


Рисунок 3.3 — Відмінності в способах віртуалізації

Застосунок, запущений в контейнері “вважає”, що він єдиний зі всією операційною системою. Ізоляція можлива за рахунок використання таких механізмів у Linux, як простори імен (namespaces) та контрольних груп (control group). Інакше кажучи, простори імен забезпечують ізоляцію в рамках операційної системи, а контрольні групи встановлюють обмеження на споживання контейнером ресурсів фізичної машини, щоб зробити рівномірним розподіл навантаження між запущеними контейнерами [26].

Тобто контейнери самі по собі не є новою технологією, просто проект Docker інкапсулював складні механізми просторів імен і контрольних груп, і оточений екосистемою, яка забезпечує зручне використання контейнерів на всіх стадіях розробки програмного забезпечення.

Образ можна розглядати як набір файлів. До його складу входить все необхідне для запуску і роботи застосунку на фізичній машині з встановленим Docker: операційна система, середовище виконання і застосунок який готовий до розгортання.

Але при такому розгляді виникає проблема використання декількох образів на одній фізичній машині, і буде недоцільним з точки зору навантаження і зберігання, щоб кожний образ загрузав все необхідне для своєї роботи, адже більшість файлів

будуть повторюватися, а відрізнятися тільки файли застосунків. Структура образу дозволяє уникнути такого дублювання.

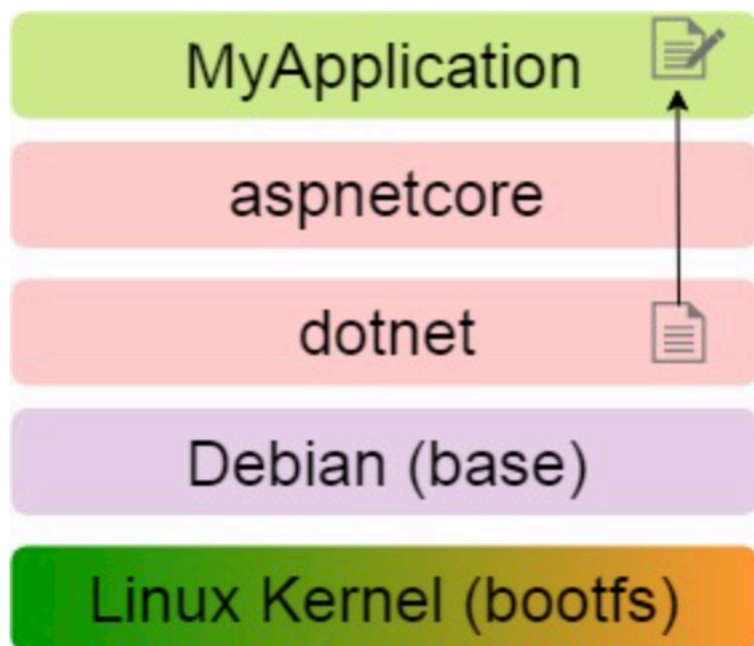


Рисунок 3.4 — приклад структури образів Docker

Образ складається з шарів (рисунок 3.4), кожний з яких представляє деяку незмінювану файлову систему. Образ в цілому представляє собою об'єднану файлову систему (Union File System), яку можна розглядати як результат поєднання шарів різних файлових систем. Об'єднана файлова система вміє опрацьовувати конфлікти, наприклад коли в різних шарах присутні файли і директорії з однаковими іменами.

Технологію Docker було вибрано для ефективного використання апаратних ресурсів, ізоляції та забезпечення незалежності програмного забезпечення від апаратної платформи, інкапсуляції всіх необхідних ресурсів для вирішення окремих задач використовується технологія віртуалізації. Віртуалізація забезпечує:

- гарантоване продуктивне функціонування Системи;
- централізоване керування та автоматизацію при впровадженні нових застосувань і сервісів;
- динамічне об'єднання окремих апаратних ресурсів в виділені логічні об'єми доступних ресурсів;
- керування існуючими ресурсами та масштабування;

- динамічне розподілення обчислювальних ресурсів відповідно до заданих політик;
- гарантовану відмово-стійкість в рамках визначених політик на рівні програмного забезпечення та апаратних засобів.

### 3.6. Трирівнева архітектура

Існує дуже багато видів і типів архітектур, які успішно застосовуються. Одною з найбільш поширених є класична трирівнева система, яка передбачає розділення програми на три рівні.

Варто зазначити, що багаторівневою архітектурою часто позначають два не зовсім зв'язаних поняття: *n-layer* та *n-tier*. І *layer* і *tier*, як правило, позначаються словом “рівень”, деколи по відношенню до “*layer*” ще вживають слово “шар”. Одна в багатьох випадках рівні будуть різного порядку.

*Tier* представляє фізичний рівень. Тобто, якщо йде мова про трирівневу архітектуру, то програму можна розділити на наступні рівні:

- сервер бази даних;
- веб-застосунок на веб-сервері;
- браузер користувача.

Кожний рівень представляє собою деякий фізичний процес, навіть якщо сервер бази даних, веб застосунок і браузер користувача знаходяться на одній фізичній машині. Якщо в якості клієнта використовувався б ще мобільний додаток, то це був би ще один фізичний рівень.

*Layer* представляє собою логічний рівень. Тобто можливі рівні доступу до даних, рівень бізнес-логіки, рівень презентації, рівень сервісів і т.д. При цьому логічні рівні не співпадають з фізичними. В даній системі керування галузевим реєстром, рівень презентації містить і контролери, які опрацьовують вхідні події, і представлення, які відображаються у веб-браузері, тобто розділяються на два фізичні

рівні.

В даному випадку йде мова саме про логічні рівні, тобто про n-layer архітектуру.

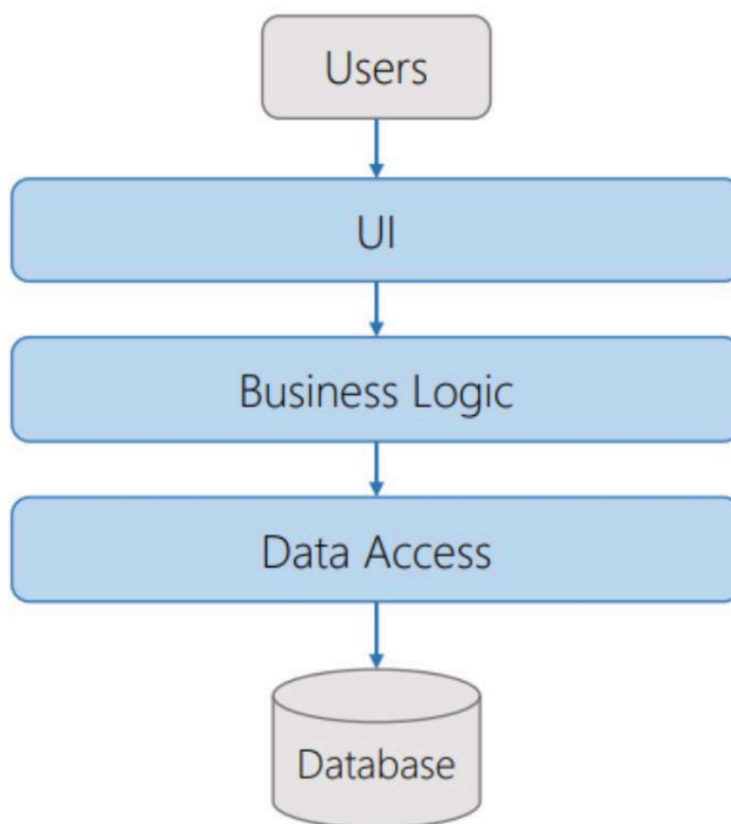


Рисунок 3.5 — візуалізація тришарової архітектури

В розробленій системі трирівнева архітектура складається з наступних рівнів (рисунок 3.5):

- Presentation layer (рівень презентацій) – рівень, елементи якого формують користувацький інтерфейс (стили, статичні сторінки html, javascript), а також моделі представлень, контролери і об’єкти контексту запиту.
- Business layer (рівень бізнес логіки) – містить набір компонентів, які відповідають за опрацювання даних, отриманих від рівня представлень. Варто також додати, що даний рівень реалізує всю необхідну логіку застосунку, всі обчислення, взаємодіє з базою даних, і передає рівню презентацій результат обробки.
- Data Access Layer (рівень доступу до даних) – зберігає моделі сутності бази даних. Також тут розташовуються різні класи, які специфічні для роботи з різними

технологіями доступу до даних. Тут також зберігаються репозиторії, через які рівень бізнес-логіки взаємодіє з базою даних.

При цьому потрібно враховувати, що крайні рівні не можуть взаємодіяти між собою, тобто рівень презентацій не може напряду звертатись до бази даних і навіть до рівня доступу до даних, а тільки через рівень бізнес-логіки.

Рівень доступу до даних не залежить від інших рівнів, на відміну від рівня бізнес-логіки, який залежить від попереднього, і рівня презентацій, який залежить від рівня бізнес-логіки.

Компоненти, як правило, повинні бути слабко-зв'язаними, тому невід'ємною частиною багаторівневих застосунків є патерн впровадження залежностей (dependency injection).

### **3.7 Сервіс балансування NGINX**

Сервіс балансування NGINX являє собою вільний веб-сервер і проксі-сервер. Основні його функції, як HTTP-сервера полягають в наступному:

- кешування статичного контенту;
- кешування HTTP-запитів;
- відмостійкість при збільшенні навантаження;
- фільтрація запитів;
- підтримка SSL.

В NGINX дуже важливим є поняття віртуальних серверів. Конфігурація дозволяє визначити віртуальні сервери, які фізично знаходяться і обслуговуються одним сервером. Віртуальні сервери поділяються на локації.

Для ефективного керування пам'яттю NGINX використовує пули – послідовність попередньо виділених блоків динамічної пам'яті.

Важливо відзначити модуль в Nginx, який відповідає за географічну класифікацію клієнтів за IP адресою. Його основу складає база даних асоціацій адрес



географічного регіона, представлена у вигляді префіксного дерева.

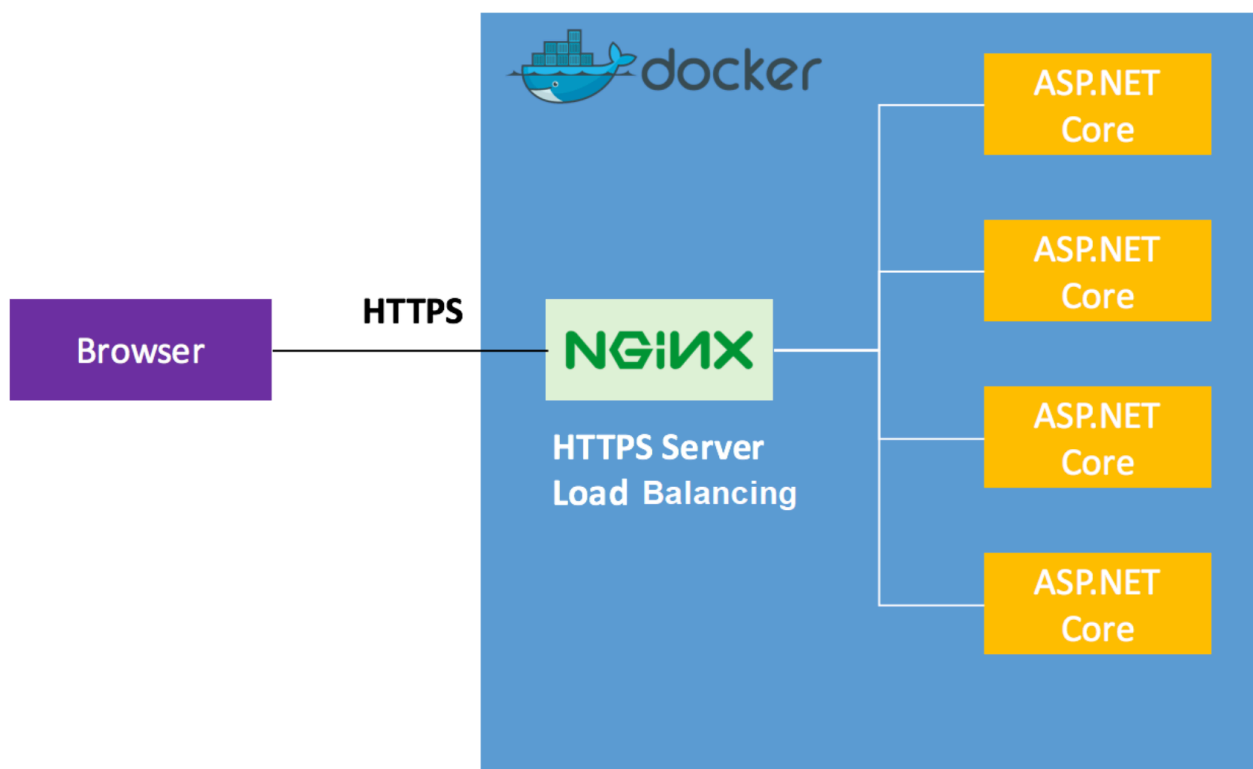


Рисунок 3.6 — візуалізація Nginx серверу в архітектурі розробленої системи

На ілюстрації вище (рисунок 3.6) можна побачити візуалізацію ролі серверу балансування навантаження Nginx, в архітектурі розробленої системи. Підсумовуючи, даний сервер є першою ланкою оброблення користувацьких запитів, шифруючи трафік з користувачем в відкритій мережі по HTTPS, і балансуючи навантаженнями між декількома розгорнутими серверами управління галузевим реєстром. Варто також зазначити, що сервер Nginx, як і будь-яка інша інфраструктурна одиниця запускається в Docker контейнері.

### 3.8. Висновки до розділу 3

Вибираючи стек технологій для реалізації системи управління галузевим реєстром вирішальним фактором було забезпечення крос-платформності та наявності

бібліотек і фреймворків з відкритим вихідним кодом та готовими рішеннями, з метою спрощення реалізації. Незалежність від операційних систем і платформ досягається шляхом використання технологій віртуалізації Docker та docker-compose. Платформа NET Core підтримує інтеграцію з технологією Docker, тому була вибрана для реалізації системи.

## **4. ДЕКОМПОЗИЦІЯ СИСТЕМИ ТА ОСОБЛИВОСТІ АРХІТЕКТУРИ**

Для візуалізації архітектури розробленої системи були використані UML та діаграми типу C4. Їх концепція полягає в тому, що з допомогою даних візуалізацій можна показати статичну структуру програмної системи з точки зору контейнерів, компонентів та коду. Таким чином можна підібрати єдину мову, як спосіб для візуалізації архітектури програмних систем.

Програмна система є найвищим рівнем абстракції і описує щось, що несе певну цінність для користувачів.

Контейнером можна вважати сутність яка зберігає програмний код або містить дані. Іншими словами це інфраструктурна одиниця системи, яку потрібно розгортати, для того щоб система працювала. Прикладом може бути система управління базами даних, веб-застосунок, веб-сервер тощо.

Компонент являє собою угруповання пов'язаних функціональних можливостей, укладених за чітко визначеним інтерфейсом. Важливо зазначити, що всі компоненти всередині контейнера, як правило, виконуються в одному і тому ж просторі процесу.

### **4.1. Опис програмної системи на прикладі діаграми прецедентів**

В системі реалізовано наступні ролі користувачів реєстру:

- адміністратор;
- автор ресурсу;
- користувач.

Роль адміністратора полягає в тому, щоб підтверджувати реєстрацію користувачів, ресурсів, а також керувати каталогами та налаштовувати користувацькі групи.

Автор ресурсу має змогу здійснювати навігацію по реєстру, пошук і отримання ресурсу відповідно до свого рівня доступу. Дана роль потрібна для історичної довідки

та аудиту. Автором може виступати фізична особа або організація, в якій є акаунт в даній системі.

Звичайний користувач може здійснювати навігацію по реєстру, робити пошук і переглядати ресурс, якщо дозволи користувача дозволяють це зробити.

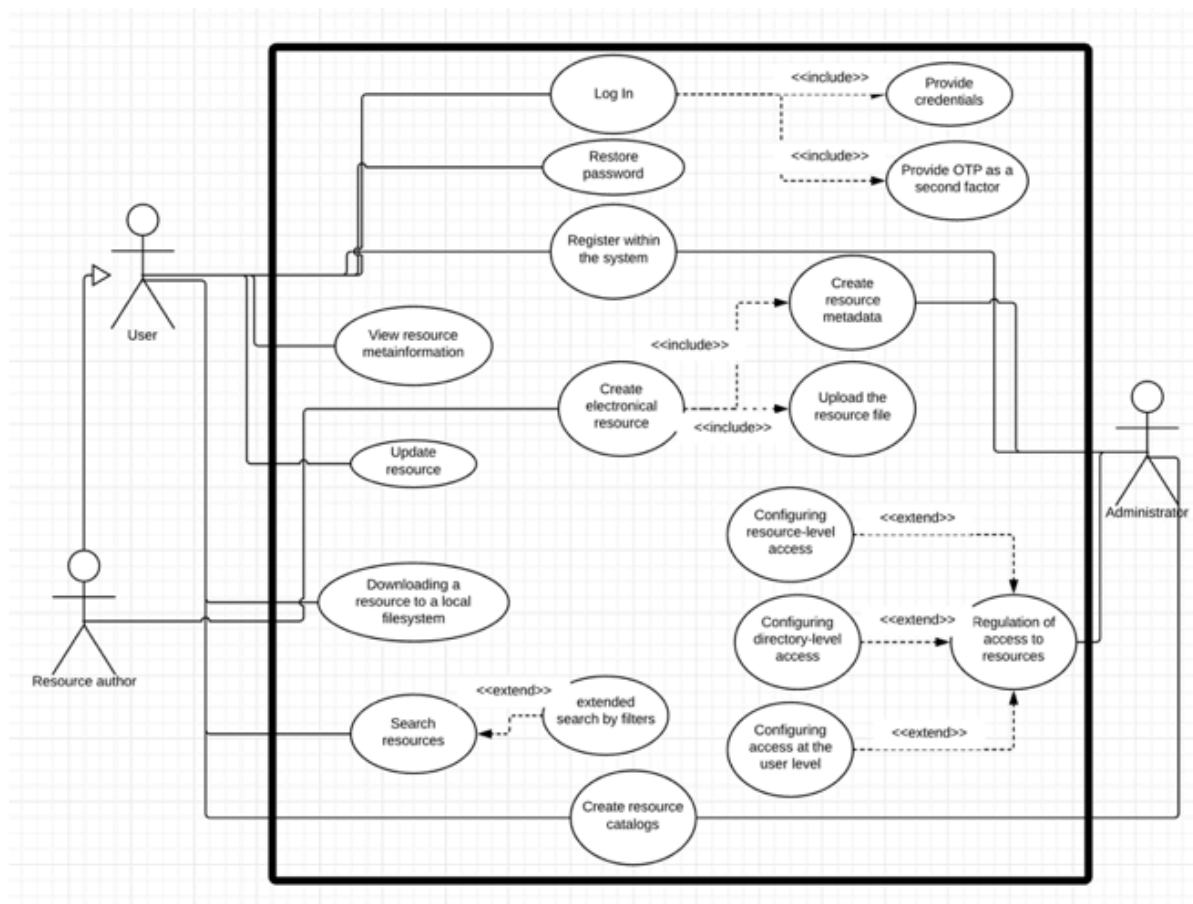


Рисунок 4.1. Діаграма прецедентів системи управління галузевим реєстром

Як видно з діаграми, наведеної вище, роль користувача і автора майже нічим не відрізняється, за виключенням того, що остання використовується для того, щоб фіксувати інформацію про дії, виконані над електронними ресурсами в реєстрі.

## 4.2. Взаємодія програмної системи із зовнішніми сервісами

Крім взаємодії з користувачами і адміністраторами, система використовує функціонал зовнішніх сервісів від Amazon – SNS, та EMAIL і SMS subscribers.

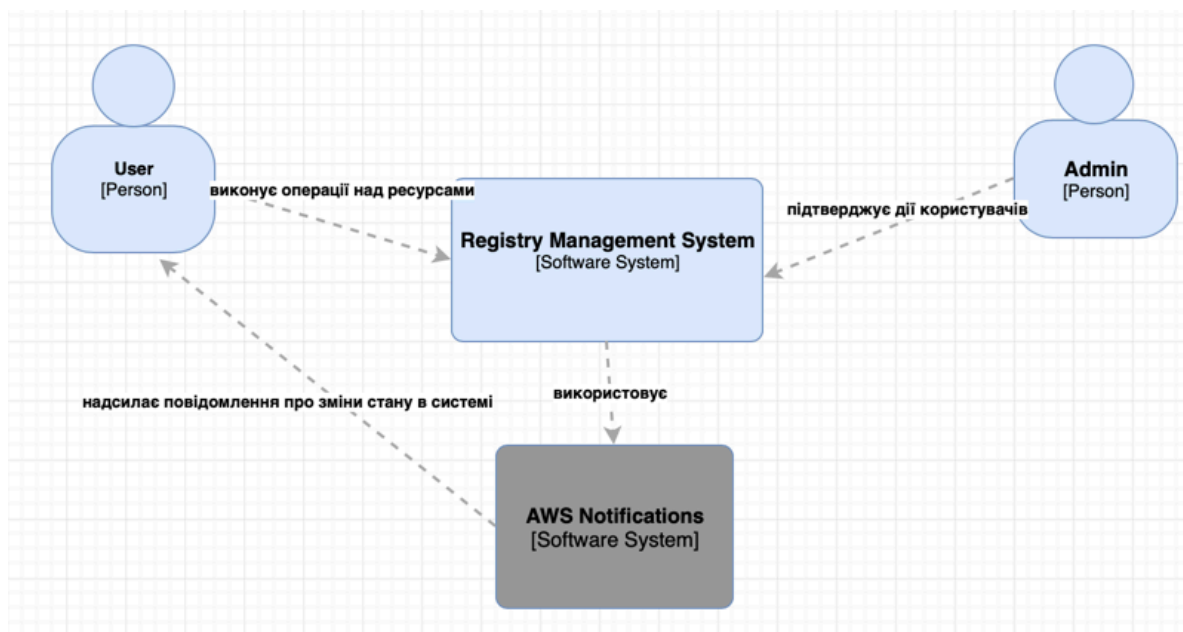


Рисунок 4.2. Глобальна діаграма архітектури системи

З діаграми наведеної на Рис. 4.2 можна побачити, що Amazon SNS використовується для асинхронного сповіщення користувачів через різні канали зв'язку.

### 4.3. Відокремлення інфраструктурних одиниць архітектури

В системі управління галузевим реєстром були відокремлені наступні інфраструктурні одиниці:

- веб-застосунок;
- сервіс балансування NGINX;
- веб-сервер ASP NET Core;
- реляційна СУБД Microsoft SQL Server.

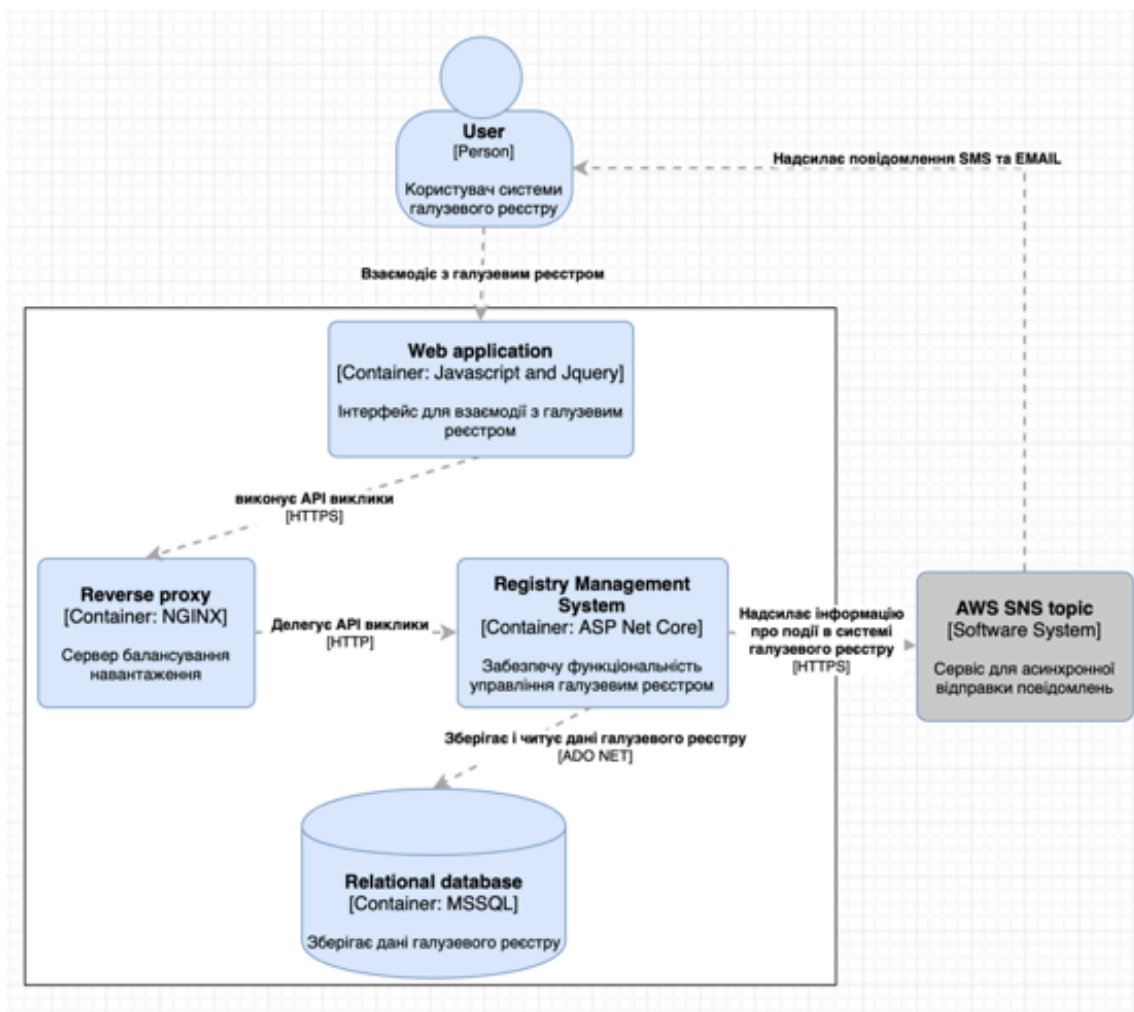


Рисунок 4.3. Діаграма архітектури веб-системи управлінням галузевим реєстром

На Рис. 4.3 відображені між інфраструктурними елементами. Важливим буде відзначити те, що протокол обміну інформацією між веб-застосунком і веб-сервером, проходить по захищеному каналі зв'язку HTTPS, через сервер балансування NGINX.

#### 4.4. Відокремлення основних модулів в програмній реалізації

В рамках веб-серверу ASP NET Core, в якому відбувається керування галузевим реєстром були виділені наступні компоненти:

- Resource Controller – компонент, який містить в собі в собі функціональність керування станом електронних ресурсів;

- Search Controller – компонент, який надає можливості пошуку та ідентифікації електронних ресурсів;
- Sign In Controller – компонент, який містить функціональність аутентифікації та авторизації користувачів системи;
- Profile Controller – компонент, який надає можливість виконувати дії керування над профілем користувачів;
- Unit of work – компонент, який інкапсулює роботу з базою даних;
- Notification component – компонент, який інкапсулює в собі функціональність сповіщень користувачів через обраний канал зв'язку.

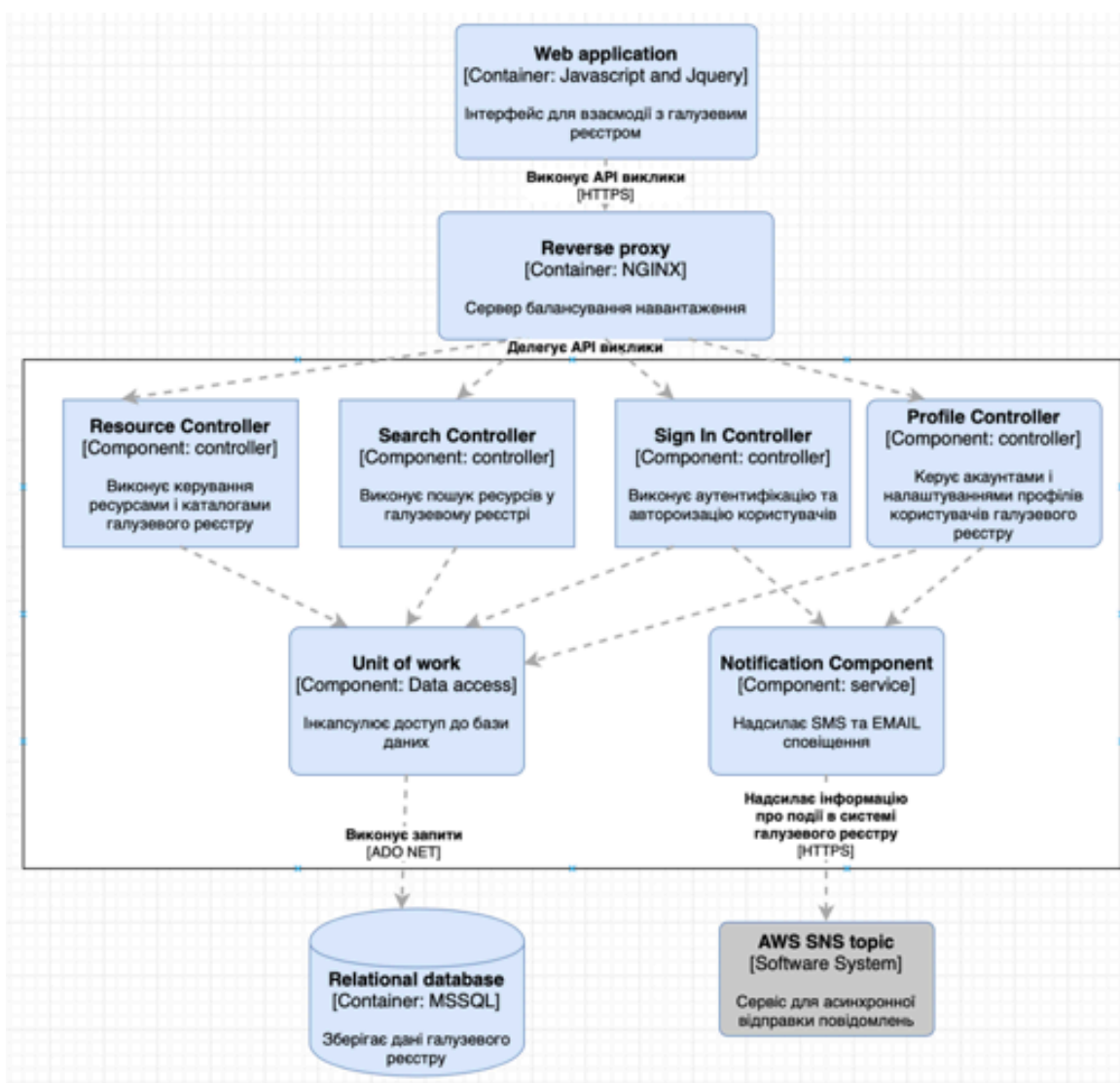


Рисунок 4.4. Діаграма компонентів системи

На Рис 4.4 проілюстровано як модулі веб серверу взаємодіють з іншими інфраструктурними одиницями системи.

## 4.5. Опис схеми бази даних

Головною сутністю в базі даних є таблиця “Resources” (рисунок 4.5). Умовно атрибути сутності можна класифікувати на три типи: загальні атрибути, пошукові атрибути ресурсу і технічні атрибути ресурсу.
















Resources	
 <b>Id</b>	bigint
 <b>Title</b>	nvarchar(100)
 <b>Description</b>	nvarchar(400)
 <b>Language</b>	nvarchar(50)
 <b>Format</b>	nvarchar(50)
 <b>SecurityLevel</b>	int
 <b>Priority</b>	int
 <b>ResourceStatus</b>	int
 <b>Location</b>	nvarchar(1000)
 <b>CatalogId</b>	bigint
 <b>IsEditable</b>	bit
 <b>FileName</b>	nvarchar(50)
 <b>CreationTimestamp</b>	datetime
 <b>ApprovalTimestamp</b>	datetime
 <b>AuthorId</b>	nvarchar(450)

Рисунок 4.5 — Табличка Resources

Загальні атрибути складаються з наступних полів:

- назва ресурсу (Title);
- опис(Description);
- дата створення ресурсу(CreationTimestamp);
- дата затвердження ресурсу(ApprovalTimestamp).

Пошукові атрибути являють собою деяку інформацію, по якій здійснюється пошук та ідентифікація електронних ресурсів.

Пошукові атрибути складають з наступних полів:

- ключові слова;



- період дати створення ресурсу;
- період дати затвердження ресурсу;
- автор(Author);
- ім'я файлу ресурсу(FileName);
- назва ресурсу(Title);
- опис ресурсу(Description).

Варто відзначити, що такі поля як назва і опис ресурсу можна віднести як і до загальних, так і до пошукових атрибутів.

Технічні атрибути ресурсу складаються з наступних полів:

- формат файлу ресурсу;
- можливість змінювати ресурс;
- розмір ресурсу;
- ім'я файлу ресурсу.

Іншою важливою сутністю в системі є таблиця “Users” (рисунок 4.6).

Основними полями є:

- ідентифікатор (Id);
- логін користувача(UserName);
- адреса електронної пошти(Email);
- номер телефону(PhoneNumber);
- рівень доступу користувача в системі(SecurityLevel).



















AspNetUsers	
 <b>Id</b>	nvarchar(450)
 <b>UserName</b>	nvarchar(256)
 <b>NormalizedUserName</b>	nvarchar(256)
 <b>Email</b>	nvarchar(256)
 <b>NormalizedEmail</b>	nvarchar(256)
 <b>EmailConfirmed</b>	bit
 <b>PasswordHash</b>	nvarchar(max)
 <b>SecurityStamp</b>	nvarchar(max)
 <b>ConcurrencyStamp</b>	nvarchar(max)
 <b>PhoneNumber</b>	nvarchar(max)
 <b>PhoneNumberConfirmed</b>	bit
 <b>TwoFactorEnabled</b>	bit
 <b>LockoutEnd</b>	datetimeoffset
 <b>LockoutEnabled</b>	bit
 <b>AccessFailedCount</b>	int
 <b>AccountStatus</b>	int
 <b>UserGroupId</b>	bigint
 <b>SecurityLevel</b>	int

Рисунок 4.6 — Табличка Users

Також варто відзначити деякі службові атрибути, які в цілому покращують роботу системи:

- **NormalizedUserName** і **NormalizedEmail** – модифікована версія полів **UserName** та **Email**, які створені з метою пришвидшення роботи запитів до бази даних, шляхом створення індексів на ці поля, що дасть змогу швидко виконувати такі операції як перевірка на унікальність незалежно від реєстру;

- **Мітка безпеки (SecurityStamp)** – поле, яке містить хеш код, який складається з авторизаційної інформації користувача (ім'я, пароль, електронна пошта і т.д.). Коли така інформація змінюється, разом з нею перераховується поле **SecurityStamp**, що дає змогу швидко визначити момент часу, коли потрібно авторизувати користувача ще раз. Дана мітка була створена з міркувань безпеки;

- **Багатопотокова мітка (ConcurrencyStamp)** – поле, яке містить випадково-

згенерований рядок, вміст якого перевіряється перед збереженням змін в базі даних. Це потрібно для оптимістичного блокування, щоб уникнути ситуації, де користувачі системи можуть перетерти зміни один одного при одночасній роботі з електронним ресурсом [17].

Таблиця Catalogs (рисунок 4.7) — є деякою композитною одиницею, яка може об'єднувати в собі інші каталоги або ресурси





Catalogs	
 <b>Id</b>	bigint
 <b>Name</b>	nvarchar(100)
 <b>SecurityLevel</b>	int
 <b>SuperCatalogId</b>	bigint

Рисунок 4.7 — Табличка Catalogs

Варто також відзначити таблицю EFMigrationHistory, яка містить в собі інформацію про послідовні зміни (міграції) структури таблицю в базі даних галузевого реєстру.



__EFMigrationsHistory	
 <b>MigrationId</b>	nvarchar(150)
 <b>ProductVersion</b>	nvarchar(32)

Рисунок 4.8 — Табличка EFMigrationHistory

В ній варто відзначити наступні поля:

- ідентифікатор міграції(MigrationId) – унікальний ідентифікатор міграції;
- версія програми(ProductVersion) – версія програми, в якій була проведена

дана міграція

## 4.6. Висновки до розділу 4

Архітектура програмного продукту розроблялася з огляду на необхідність подальшої підтримки, і облегшення експлуатації. Контейнеризація та віртуалізація в

цілому вирішують другу проблему, і роблять програму крос-платформною. Саме тому була вибрана платформа NET Core, яка підтримує контейнеризацію Docker. Також варто відзначити, що сервіс відправки асинхронних повідомлень AWS SNS доповнив існуючу архітектуру, використовуючи підходи патерну “event-driven” [9].

## 5. РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

### 5.1. Вимоги для функціонування системи

Для нормального розгортання, даній системі необхідна попередня установка технологій віртуалізації Docker та docker-compose.

Всі інші середовища виконання та бібліотеки будуть розгортатись безпосередньо в Docker контейнері, полегшуючи при цьому простоту використання і експлуатації програмного продукту.

### 5.2 Інструкція для роботи з системою

Після запуску програми з'являється домашня сторінка галузевого міністерства. (рисунок 5.1).

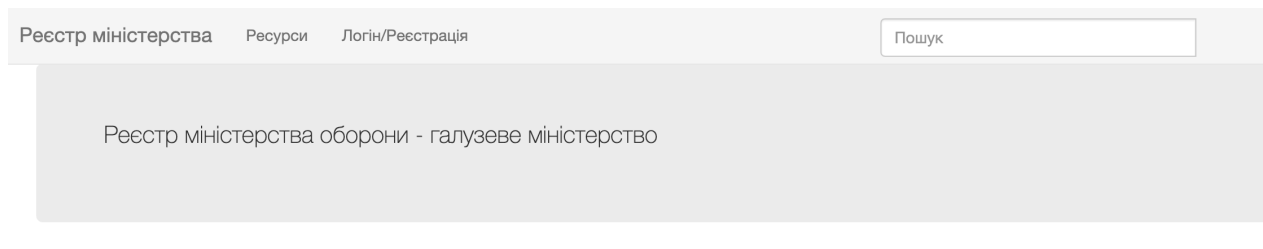


Рисунок 5.1 – Домашня сторінка галузевого міністерства

Після цього користувачу необхідно пройти етап двох-факторної автентифікації (якщо вона була включена користувачем раніше) та авторизації, як показано на (рисунок 5.2).

Реєстр міністерства
Ресурси
Логін/Реєстрація

Пошук

## Логін.

Використайте свій акаунт для входу

Ім'я користувача

Пароль

☒ Запам'ятати мене?

Ввійти

[Зареєструватися](#)  
[відновити пароль](#)

Рисунок 5.2 – Перший етап автентифікації

Після чого буде згенерований одноразовий пароль і надісланий на телефон користувача для підтвердження особистості. З ілюстрації, наведеної вище видно, що у користувача також є можливість зареєструватись у системі та відновити пароль.

Після проходження автентифікації та авторизації користувач знов побачить домашню сторінку, тільки уже з різним набором функціоналу, в залежності від ролі (адміністратор або звичайний користувач).

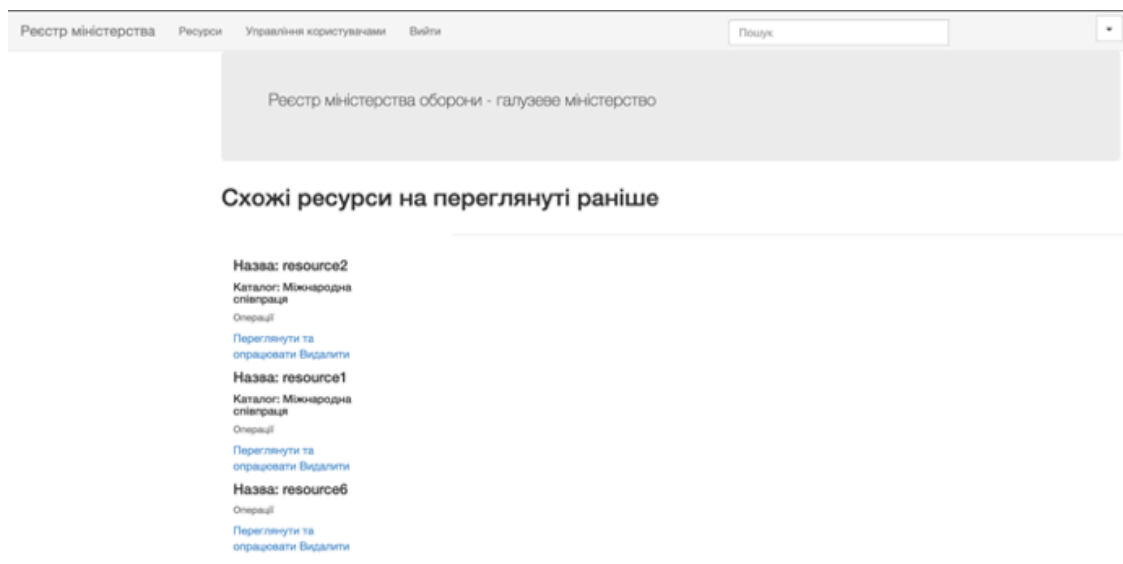


Рисунок 5.3 – домашня сторінка адміністратора

На ілюстрації вище (рисунок 5.3) зображені можливості адміністратора в системі. Він має можливість переглянути ресурси, схожі з тими, які він переглядав раніше, здійснювати керування над користувачами, а саме

- підтверджувати реєстрацію користувачів;
- змінювати роль користувачів.

#### Користувачі

UserName	AccountStatus	Email	PhoneNumber	AccessFailedCount	
JohnDoe	Approved	test_user@test.com		0	<a href="#">Редагувати</a>   <a href="#">Деталі</a>   <a href="#">Видалити</a>
JimGarham	НЕПІДТВЕРДЖЕНИЙ	test_user2@test.com		0	<a href="#">Редагувати</a>   <a href="#">Деталі</a>   <a href="#">Видалити</a>

Рисунок 5.4 – керування користувачами

На малюнку вище (рисунок 5.4) зображений інтерфейс для управління користувачами.

Користувач має змогу керувати реєстром (в залежності від рівня доступу в системі). Керування електронними ресурсами передбачає наступні дії:

- створення нового електронного ресурсу;
- редагування атрибутів ресурсу;
- видалення ресурсу;
- перегляд вмісту ресурсу;
- шукати ресурс за його атрибутами.

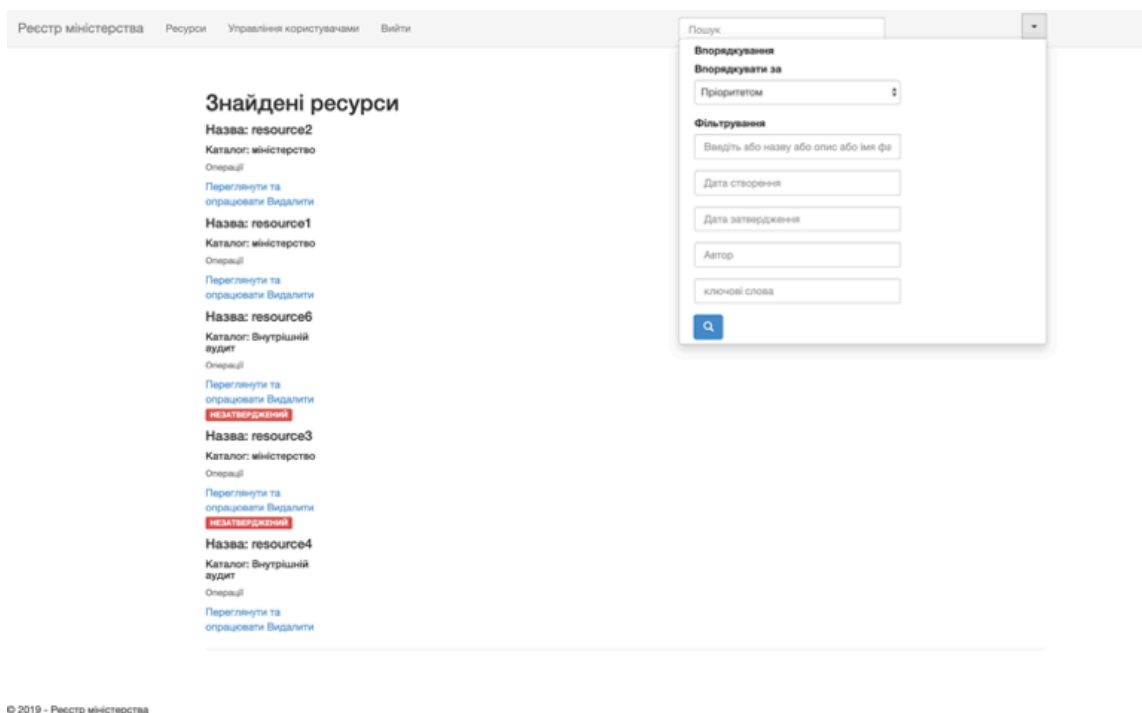


Рисунок 5.5 – можливості розширеного пошуку

На ілюстрації вище (рисунок 5.5) зображені можливості розширеного пошуку в системі управління галузевими реєстрами.

Варто зазначити, що адміністратор теж може виконувати керування над реєстром, а також підтверджувати зміни користувача над електронними ресурсами (рисунок 5.6).



Реєстр міністерства
Ресурси
Управління користувачами
Вийти
Пошук

### Визначити атрибути ресурсу

Атрибути

Title	resource2
Description	description
Language	ukr
CreationTimestamp	11/11/2019, 18:17:25
ApprovalTimestamp	11/17/2019, 00:10:21.373
Format	.txt
Author	
SecurityLevel	5
Catalog	міністерство
Priority	5
ResourceStatus	Approved
Tags	tag4 tag9

Зберегти
Завантажити документ
Дозволити редагування
Підтвердити
Видалити ресурс

[Повернутись до реєстру](#)

© 2019 - Реєстр міністерства

Рисунок 5.6 – керування ресурсом

Також варто відзначити, що адміністратор має можливість дозволити редагування електронного ресурсу, який по замовчуванню є незмінним.

### 5.3. Висновки до розділу 5

Користувацький інтерфейс розроблявся з метою зробити всі дії користувачів інтуїтивно зрозумілими, а також забезпечити узгодженість стану галузевого реєстру. Це досягається шляхом підтвердження адміністратором всіх дій користувачів, які змінюють стан ресурсів.

Також варто зазначити, що в системі фіксуються всі дії, які користувачі роблять з реєстром. Це дає змогу швидко ідентифікувати зловживання рівнем доступу, а також використання секретної інформації електронних ресурсів у заборонений спосіб.

## 6. РОЗРОБКА СТАРТАП ПРОЕКТУ

Стартап – це нещодавно створена компанія, яка знаходиться лише на стадії розвитку, але має інноваційні ідеї. Саме ці нововведення допоможуть перевершити конкурентів та вийти на ринок з новим продуктом, послугою. Основна особливість стартапів – недостатність фінансів, пошук «бізнес-ангелів» та відношення не тільки до ІТ-сфери, а і до будь-якої області ринку. Зараз стартап-проекти можна називати венчурними.

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту. В межах цього етапу:

- 1) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- 2) аналізуються ринкові можливості щодо його реалізації;

3) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

## 6.1 Опис ідеї проекту

Далі було проаналізовано і подано у вигляді таблиць:

- 1) зміст ідеї, що пропонується;
- 2) можливі напрямки застосування;
- 3) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- 4) відмінність від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 6.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 6.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблену систему для управління галузевими реєстрами можна впровадити в компанії з різними моделями роботи як рішення, яке автоматизує документообіг	1. Використання системи як готового рішення в новій галузі	Можливість отримати готове рішення для автоматизації документообігу прикладаючи мінімум зусиль
	2. При проектуванні нової системи управління реєстрами	В даному випадку користувачем буде повна компанія. Вигода буде полягати в можливості перейняти архітектурне чи технологічне рішення при проектуванні нової системи

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

1) визначення переліку техніко-економічних властивостей та характеристик ідеї;

2) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

3) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 6.2).

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик проекту

№ п/п	Техніко- економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтр. сторона)	S (сильна сторона)
		Мій проект	GSA REGISTER			

продовження таблиці

1.	Персоналізація	Використання інформації про переглянуті раніше ресурси, для рекомендації нових	Можливості персоналізації відсутні	-		
2.	Адаптивність до будь-якої галузі	Наявна можливість адаптації до різних галузей завдяки динамічній схемі метаінформації ресурсів	Можливість використання тільки в бібліотечній галузі			+
3.	Можливість розширеного пошуку та класифікація документів	Можливість здійснювати пошук по багатьох атрибутах ресурсів	Присутня можливість робити базовий пошук			+
4.	Зручність використання/орієнтованість на користувача	Зручний інтерфейс з підтримкою візуалізації	Зручний ін.-терфейс для управління		+	

## 6.2 Технологічний аудит ідеї продукту

Було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.3):

- 1) за якою технологією буде виготовлено товар згідно ідеї проекту?
- 2) чи існують такі технології, чи їх потрібно розробити/доробити?
- 3) чи доступні такі технології авторам проекту?

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як ASP NET Core на платформі NET Core через їх доступність та безкоштовність. Також було використано сервер балансування NGINX для забезпечення висконавантажуваності системи.

Таблиця 6.3 – Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення програмного забезпечення для управління галузевими реєстрами електронних інформаційних ресурсів	Алгоритми рекомендації та пошуку електронних ресурсів	Наявна	Доступна
		Розподілення навантаження на сервер	Наявна	Доступна
		jQuery/Bootstrap	Наявна	Доступна
Обрані технології реалізації ідеї проекту: алгоритми рекомендації та пошуку + розподілення навантаження + jQuery/Bootstrap				

### 6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	420000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність вхідних обмежень (вказати характер обмежень)	-
№ п/п	Показники стану ринку (найменування)	Характеристика
5	Специфічні вимоги до стандартизації та сертифікації	-
6	Середня норма рентабельності в галузі (або по ринку), %	49

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 6.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та зформовано орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Програмне забезпечення для управління галузевими реєстрами електронних інформаційних ресурсів	Архітектори систем, які автоматизують документообіг електронних ресурсів	Мета використання програмного забезпечення: для автоматизації обліку електронними документами і масивами документів	Зручний інтерфейс, швидкість роботи системи, а також хороші механізми класифікації та ідентифікації електронних ресурсів

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 6.6, 6.7).

Таблиця 6.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	<p>Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів.</p> <p>Вдосконалення технічних моментів і вибір цільової аудиторії.</p>



продовження таблиці

2	Невідповідність умовам технологічного розвитку	Динамічний розвиток технологій, що призведе невідповідності ПП використовуваним у сучасних приміщеннях технологіям	Забезпечення гнучкості та рекомендаційних класифікаційних моделей, адаптація до сучасних умов швидкими темпами
3	Зміна потреб користувачів	Користувачам необхідним програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створеного ПП

Таблиця 6.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для вітчизняного користувача.	Адаптація програмного продукту до вітчизняних особливостей.
№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
2	Поява нових методів класифікації та рекомендації	З'являться нові методи і підходи для управління галузевими реєстрами	Покращити ПП додаванням нового функціоналу, розширення можливостей

Надалі було проведено аналіз пропозиції: визначено загальні риси конкуренції на ринку (таблиця 6.8).

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Конкуренція: чиста	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні нововведення, вдосконалення.
2. Рівень конкурентної боротьби: інтерціональний	Компанії-конкуренти з різних країн	Створити ПП з урахуванням потреб користувачів з різних країн світу
3. Галузева ознака: незалежить від галузі	Продукт може використовуватись у різних галузях	Постійне вдосконалення продукту з метою його застосовності у інших сферах
4. Конкуренція за видами товарів: товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недоліки конкурентів
5. Характер конкурентних переваг: неціновий	Вдосконалення технології створення ПП, для зниження ціни і розширення функціональності	Удосконалення моделі. Використання більш дешевих тех.нологій для розробки, ніж використовують конкуренти, але тільки за їх відповідності вимогам якості.
6. Інтенсивність: не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (таблиця 6.9).

За результатами аналізу таблиці 6.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок

щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок було враховано при формулюванні переліку факторів конкурентоспроможності далі. На основі аналізу конкуренції, наведеного в таблиці 6.9, а також з урахуванням характеристик ідеї проекту (таблиця 6.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиці 6.6, 6.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у таблиці 6.10.

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	OCLC, FED LINK, GSA REGISTER	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, авторитет (перевірена якість)

продовження таблиці

Висновки:	Досить інтенсивна конкурентна боротьба з гравцями, що вже закріпилися на ринку	Є можливості виходу на ринок, але є і конкуренти. Приблизний термін – 6 місяців.	-	Зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для управління галузевими реєстрами	Необхідно випустити ПЗ на рівні з конкурентами та розширювати функціонал.
-----------	--	--	---	--	---

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (перелік чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Доступніша ціна збільшує кількість потенційних клієнтів
№ п/п	Фактор конкурентоспроможності	Обґрунтування (перелік чинників, що роблять фактор для порівняння конкурентних проектів значущим)
2	Орієнтованість на кінцевого користувача	Продукт орієнтований на взаємодію з клієнтом
3	Адаптованість під конструкторські особливості конкретного приміщення	Ширше коло будинків, для яких може бути використаний ПП

За визначеними факторами конкурентоспроможності (таблиця 6.10) проведено аналіз сильних та слабких сторін стартап-проекту (таблиця 6.11).

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним проектом						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	13			+				
2	Орієнтованість на кінцевого користувача	10				+			
3	Адаптованість під конструкторські особливості конкретного приміщення	20		+					

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 6.12 – SWOT-аналіз стартап-проекту

<u>Сильні сторони:</u> Ціна Орієнтованість на кінцевого користувача	<u>Слабкі сторони:</u> Адаптованість під особливості конкретного супутника
---	---

продовження таблиці

<u>Можливості:</u>	<u>Загрози:</u>
Конкуренція	Невідповідність умовам технологічного розвитку
Поява нових методів розрахунку	Зміна потреб користувачів
Поява нових методів моделювання динаміки запуску супутників	

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 6.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 6.13).

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Терміни реалізації
1	Безкоштовне розповсюдження створеного ПП	35%	12 місяців
2	Створення ПП з подальшим розповсюдженням за певну оплату	80%	18 місяців
№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Терміни реалізації
3	Створення вебсайту, на якому можна буде користуватися ПП	70%	16 місяців

Для подальшої реалізації було обрано альтернативу №2.

## 6.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 6.14).

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Практично будь-яка компанія, якій потрібно керувати документообігом	Висока	Помірний	Сильна	Середній рівень

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано товар та визначено стратегію охоплення ринку – стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
----------	--------------------------------------	---------------------------	--	---------------------------

продовження таблиці

1	Створення ПП з подальшим розповсюдженням за певну оплату	Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), орієнтованість на кінцевого користувача	Стратегія диференціації
---	--	--	---	-------------------------

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 6.16).

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Можливі обидва варіанти	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 6.5), а також в залежності від обраної базової стратегії розвитку (таблиця 6.15) та стратегії конкурентної поведінки (таблиця 6.16) розроблено стратегію позиціонування (таблиця 6.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.



Таблиця 6.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту.
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для моніторингу запуску супутників	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

В результаті отримано узгоджену систему рішень щодо ринкової поведінки стартап-компанії.

## 6.5 Розробка маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 6.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
-------	---------	----------------------------	--

продовження таблиці

1	Зручність за-стосування	Не вимагається специфічних знань для використання ПП	Використання ПП вимагає від користувача лише коректного знання прикладної галузі
2	Практичність результату	Підвищення економічності використання систем управління галузевими реєстрами	Підвищення економічності використання систем управління галузевими реєстрами електронних інформаційних ресурсів
3	Незалежність від галузі	Гнучкість використання для різних галузей	ПП не має прив'язки до конкретного типу галузей і може бути застосованим для широкого кола таких

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 6.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і/або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язане з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє собою рішення того, як буде реалізовано товар на ринку; включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості, доставка, умови оплати тощо).

Таблиця 6.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Зменшення витрат на ручну обробку документів за рахунок використання готового рішення для керування певним галузевим реєстром
II. Товар у реальному виконанні	1. Функція розширеного пошуку електронних ресурсів
	2. Функція класифікації документів
	3. Функція регулювання рівня доступу до електронних ресурсів
	Якість: надійність та забезпечення безпеки даних галузевого реєстру.
III. Товар із підкріпленням	Пакування: відсутнє
	Марка: «RegistryManagementPro»
	До продажу: відсутнє
	Після продажу: технологічна підтримка та оновлення системи
Відкритий доступ до вихідного коду та моделей пошуку і класифікацій надано не буде. Продукт буде захищено патентом.	

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання шляхом патентування. Наступним кроком є визначення цінових меж (таблиця 6.20), якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів. Аналіз проведено експертним методом. В результаті проведення аналізу встановлено верхню та нижню межі встановлення ціни на товар, що надається користувачам

Таблиця 6.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	2000\$	3200\$	Перша група має порівняно вищий рівень доходів за інші	2000\$ (конфігурація ПП не відрізняється для різних груп споживачів, тому ціна не має діапазону коливання)

Наступним кроком є визначення оптимальної системи збуту (таблиця 6.21).

Таблиця 6.21 – Формування системи збуту

№ п/п	Специфіка закупі- вельної поведінки цільових клієнтів	Функції збуту, які має виконувати по- стачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – компанії, які шукають способи під-вищення ефективності управління галузевими реєстрами. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	Встановлення кон-тактів із споживача-ми і підтримання їх. Формування попиту, стимулювання збуту. Дослідницька робота зі збору маркетин-гової інформації. Доробка товару, ви-ходячи з потреб кон-кретного покупця.	Один (від виробника одразу спо-живачу)	Прямий канал збуту до спо-живача, мінімі-зувати збутові витрати розви-ток маркетин-гового спілку-вання із спожи-вачем

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 6.22).

Таблиця 6.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Цільові клієнти – компанії, які шукають способи підвищення ефективності моделювання запуску супутників. Вони цікавляться сучасними розробками та інноваційними рішеннями.	Конференції, інтернет-конференції, семінари, огляд професійної літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	Позиція на основі порівняння фірми з товарами конкурентів; Особливості потреб споживачів	- створення репутації фірми-виробнику; · збільшення чистого прибутку та рентабельності фірми; · збільшення потоків покупців та обсягів продажу; · стабілізація обсягів продажу в період зменшення попиту.	Концепція полягатиме у апеляції до неефективності та догоровизні використання існуючих систем документообігу без використання розробленого продукту

Результатом аналізу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

## **6.6 Висновки до розділу 6**

1. Проведено аналіз програмного продукту з позиції перспективи розвитку стартап-проекту.
2. Проведено аналіз ринку існуючих рішень.
3. Вивчено потенційні можливості та ризики розвитку бізнесу та стартап-проекту.
4. Запропоновано межі цінового формування на розроблений продукт.
5. Визначено стратегії просування стартап-проекту та маркетингову програму.

## ВИСНОВКИ

В ході виконання магістерської роботи були отримані наступні результати:

- проаналізовано існуючі приклади відомих галузевих реєстрів для управління електронними інформаційними ресурсами;
- розроблено динамічну структуру галузевого реєстру електронних інформаційних ресурсів, що дасть змогу експлуатувати дане рішення в різних галузях;
- сформовано схему метаданих галузевого реєстру;
- розроблено веб-систему для управління електронними інформаційними ресурсами;
- розроблено механізми пошуку та класифікації електронних ресурсів;
- розроблено механізм регулювання доступу до ресурсів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рудюк В.В. Класифікація електронних документів ФРН // Бібліотекознавство. Документознавство. Інформологія. Наук. Журнал / Міністерство культури і туризму України, Державна академія керівних кадрів культури і мистецтв. – Київ, 2005. – №3. – С. 26-31
2. Петров В.В. Національні інформаційні ресурси. Проблеми формування, розвитку, управління і використання // Реєстрація, зберігання і обробка даних. — 2001. — Т. 3
3. Соснін О.В. Проблеми державного управління системою національних інформаційних ресурсів з наукового потенціалу України: Монографія. – К.: Інститут держави і права ім. В.М. Корецького НАН України, 2003. – 572 с.
4. Єременко Т. В. Каталогізація ресурсів Інтернету: досвід бібліотек США // Наукові та технічні бібліотеки. — 2002. — № 5. — С. 53-68.
5. OCLC [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://www.oclc.org/en/home.html?redirect=true>.
6. Данілін А.В. Стандарти і єдина архітектура інформаційних технологій. — Microsoft Press, 2003.  
<http://www.microsoft.com/Ukraine/Government/Analytics/IntegrationTechnologies/Standards.msp>.
7. Szilvassy, Judith. Basic serials management handbook. K. G. Saur, Munchen, 1996.
8. Skaggs, Bethany Latham; Poe, Jodi Welch & Stevens, Kimberly Weatherford. One-stop shopping: A perspective on the evolution of electronic resources management. OCLC Sys. Serv., 2006, 22(3), 192-206.
9. Jewell, Timothy D. & Mitchell, Anne. Electronic resource management: The quest for systems and standarts. The Serials Librarian, 2005, 48 (1/2), 137-63.
10. Liew, Chern Li; Foo, Schubert & Chennupatti, K. R. A study of graduate student end-user's use and perception of electronic journals. Online Inf. Rev., 24(4)6 302-15.
11. Sadeh, Tamar & Ellingsen, Mark. Electronic resource management systems: The need and the realization. New Library World, 2005, 106 (5/6), 208-18.



12. SQL Server [Электронный ресурс] // Microsoft. – 2017. – Режим доступа до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017-editions>.
13. Richter J. CLR via C#. – Redmond, Washington : Microsoft Press, 2012. – С. 421. – ISBN 978-0-7356-6745-7.
14. Espoito D. Programming ASP.NET Core. – Boston: Microsoft Press, 2018. – С. 416. – ISBN 978-1-50-930441.
15. TIOBE Index [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <https://www.tiobe.com/tiobe-index/>.
16. Krasner G. A. cookbook for using the model-view controller user interface paradigm in Smalltalk-80 / G. Krasner, S. Pope // 1988 Article / G. Krasner, S. Pope. – Denville: SIGS Publications, 1988. – (Journal of Object-Oriented Programming). – С. 26–49.
17. Mostarda S. Entity Framework 4 in Action / S. Mostarda, M. Sanctis, D. Bochichio, 2011. – 576 с. (Manning)
18. Портнякин И. Эффективные пользовательские интерфейсы / Иван Портнякин. — Санкт-Петербург: Лори, 2011. — 600 с.
19. Wittig A. Amazon Web Services in action / Wittig M. // Manning Publications; 1 edition, 2015. – ISBN-10 1617292885.
20. DeJonghe D. Nginx Cookbook. – O'Really Media, Inc, 2019 – ISBN 9781492049098
21. Pollack E. Dynamic SQL. Applications. Performance and Security in Microsoft SQL Server. – Appress, 2018 – ISBN 9781484243183.
22. Campbell L. Database Reliability Engineering: Designing and Operating Resilient Database Systems. – O'Really Medua, Inc, 2017 – ISBN 978-1491925942
23. Fowler M. Analysis Patterns. – Addison-Wesley, 1997.
24. Evans E. Domain-Driven Design, Addison-Wesley, 2004.
25. Fowler M. Patterns of Enterprise Application Architecture, Addison-Wesley, 2003
26. Nickoloff J. Docker in Action. – Manning Publications, 2016. – ISBN 9781617294761
27. Morgan B. Microservices in action / Morgan B. Paulo A.P. // Manning Publications, 2018 – ISBN 9781617294457

28. Петруненко А. Оценка коммерческой привлекательности проекта [Электронный ресурс]. – Режим доступа: <http://www.techbusiness.ru/tb/archiv/number2/page01.htm>
29. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
30. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.

## ДОДАТОК А

### Публікації

Розробка агента управління попитом на енергію “інтелектуальної будівлі”

УКР.НТУУ”КПІ”\_ТЕФ\_АПЕПС\_ ТІ41161\_19М

Аркушів 8

2019

Ministry of education and science of Ukraine  
National Technical University of Ukraine “Igor  
Sikorsky Kyiv Polytechnic Institute”

Modern Science for new industries  
China-Ukraine International Symposium on Innovation and  
Technology II 2019  
Part 2. Igor Sikorsky KPI expert papers

25 June 2019

Kyiv

Kiev

Igor Sikorsky Kyiv Polytechnic Institute 2019

**Modern science for new industries:** Collection of scientific papers of the Chinese-Ukrainian International Symposium on Innovation and Technology II 2019, 25 June.2019- Part 2. Igor Sikorsky KPI expert papers. – Kyiv: Igor Sikorsky Kyiv Polytechnic Institute,\_\_\_\_\_, 2019. – 119 p.

**ISBN**

Part 2 of the collection of scientific papers of the Chinese-Ukrainian International Symposium on Innovations and Technologies “Modern Science for New Industries” presents the results of scientific research and practical developments of scientists from Igor Sikorsky KPI

**Materials are printed in the author’s edition.**

**Современная наука для новых отраслей промышленности:** сб. науч. трудов Китайско-Украинского Междунар. симпозиума по инновациям и технологиям II 2019, 25 июня 2019г. Ч. 2. – Научные труды ученых КПИ. им. Игоря Сикорского – Киев: КПИ им. Игоря Сикорского,\_\_, 2019 – 119 с.

**ISBN**

Во второй части сборника научных трудов Китайско-Украинского международного симпозиума по инновациям и технологиям приведены результаты научных исследований и практических разработок ученых КПИ им. Игоря Сикорского

**Материалы печатаются в авторской редакции.**

ISBN\_\_\_\_\_

Authors of articles,2019

Igor Sikorsky Kyiv Polytechnic Institute,2019

## Contents

New materials and prospective technologies (Bogomol I., Loboda P.)	4
Influence of thermomagnetic and thermomechanical treatment on the structure, properties and parameters of martensitic transformation in shape memory alloys (Demchenko L.)	10
Data mining and its application for telecom industry (Globa L.S.)	28
Various origin wastes utilization with simultaneous energy carriers obtaining (Golub N.B., Levzun I.I.)	52
Importance of toxicological investigations in studying of prospects of new antifungal agents (Kalashnikova L., Trush M., Hodyna D., Rogalsky S., Metelytsia L.)	57
Application of theory fuzzy logic for determining the level of innovative security of Ukraine (Karaieva N.V., Annenkov M. E.)	62
Emergency risks for innovative development (Karaieva N.V.)	68
Javascript mapping for data stored in DBMS intersystems Cache (Kovalchuk K., Mykhailova I.)	72
Unmanned aerial vehicle on solar energy (Kozel Y.)	80
Web system for managing the registry of electronic information resources (Kuzmenko I., Ivaniuk A.)	84
Analysis of existing software for the controlling of water obligation (Sydorenko Iu.V., Kryvda O.V.)	92
Department of chemical technology of composite materials (Svidersky V.A., Myronyuk O.V.)	98
Modern problems of heat power engineering (Varlamov Gennadii, Katerina Romanova, Mukhin Mikhail, Wu Zongyan, Aleksandr Chebotarev)	102

## **WEB SYSTEM FOR MANAGING THE REGISTRY OF ELECTRONICAL INFORMATION RESOURCES**

National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»

A. Ivaniuk, I. Kuzmenko

### *Formulation of the problem*

A register is considered an information and telecommunication system, which is intended for registration, accounting, accumulation, processing and storage of information about content, location, providing means to access electronic information resources and meeting the needs of legal and physical persons in information services [1].

The problem of creation, exploitation, protection of electronic information resources is extremely acute today. The rapid growth of the number of information resources used by mankind confuses scientists and software developers and forces them to actively seek the solution to the problem of improving the efficiency of information resources exploitation. The main areas of search are streamlining information arrays, developing new storage methods and technologies, accessing information, and securing information.

As the experience of the United States shows, efforts to streamline electronic information resources through their cataloging make it possible to integrate the latter with a powerful system of national library directories. Therefore, most of the forces in the cataloging is applied by libraries. The projects InterCAT, CORC, INFOMINE, CATRIONA should be highlighted [2].

OCLC (Online Computer Library Center) is an international organization that brings together over 53,000 libraries in 96 countries and aims to help libraries and other agencies organize world information resources so that knowledge seekers can find them to gain new knowledge. To do this, OCLC helps libraries share descriptions of their collections and creates collaborative frameworks for organizing new knowledge. The main activities of this organization are: Cataloging and Metadata, Collection Management, Digitization and Preservation, Electronic content, Reference, Resource Sharing, DUNS (Data Universal Numbering System) [3].

Most government-level countries have developed and continue to work on standards for the creation, development and use of information resources. In UK: Government Gateway, Government Internet, e-GIF, e-GMS (e-Government Metadata) - a standard that specifies elements for describing metadata and displaying Dublin Core, AGLS, NGDF, GILS and PRO. In Denmark: infrastructure Infostructurebase, in Sweden: Government Elink (Ge), in Australia: FedLink — government gateway and secured government intranet and so on.

The goal of this web system work was managing the registry of electronical information resources (EIR) of the university department. It was necessary to analyze the existing examples of registries of electronical information resources, to compose the structure of EIR registry and to develop a software product that would allow the following functions (Fig. 1): adding a user to the registry; user authentication

and authorization in the system; adding a resource to the registry; saving resources metadata in the registry; linking directories of resources and user groups; downloading a resource to a local filesystem; view the resource by direct link.

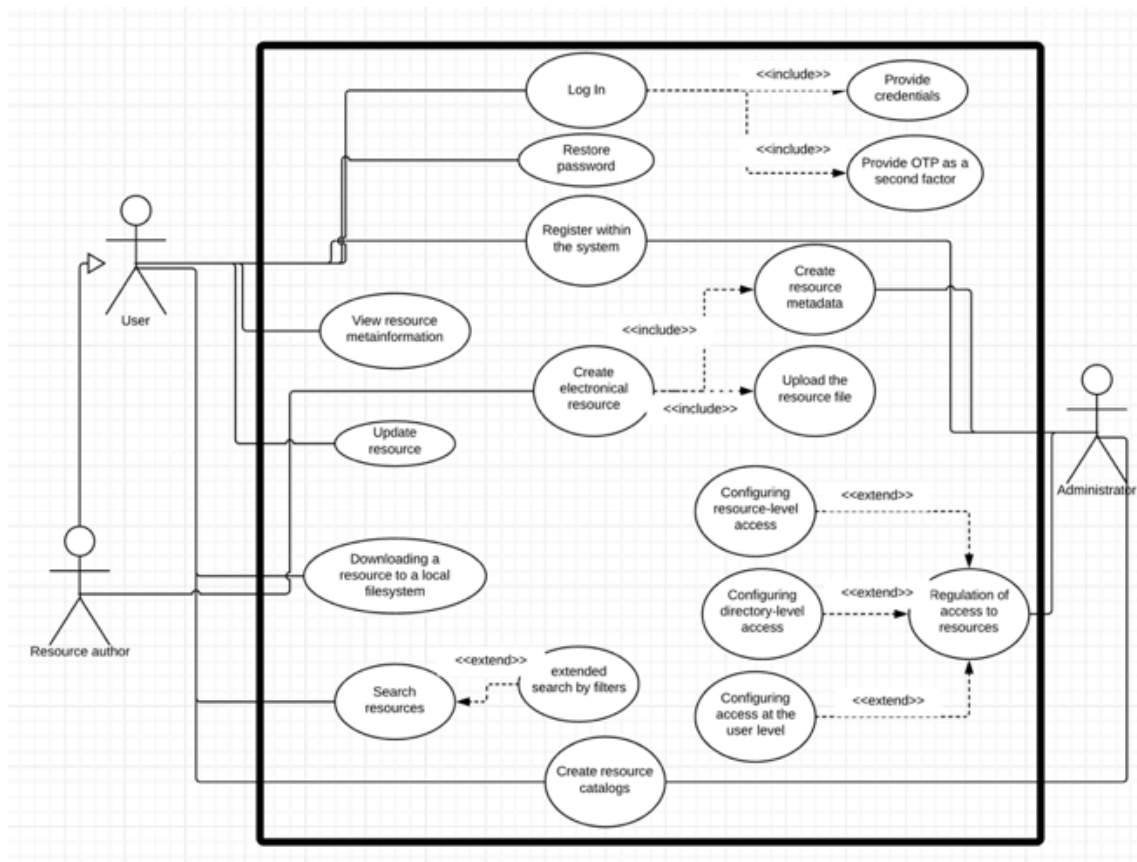


Fig. 1 – Use Case diagram of the web-system

Therefore, the purpose of the development is to create a web-based system for managing the registry of electronic information resources for departments of «Igor Sikorsky Kyiv Polytechnic Institute» which will be able to solve problems related to a single standardized access to resources, information security, as well as to be able to effectively manage the registry and integrate electronic resources by using the metadata resource schema and, as a consequence, improving their use.

### ***Description of software implementation***

After analyzing the problem statement, the main requirements for the system were identified: independent development of different parts of the software to ensure rapid functionality - this raises the need to separate the different parts of the system from each other; architectural capability to extend the functionality of the system without changing the user interface.

So, it was decided to use the modular principle of system architecture on base of object-oriented programming (OOP), which could easily solve problems with future system expansion best meets these criteria. The modular principle of system architecture has the following advantages:

- distributed system development helps to find bugs faster, unlike programs where the system interacts



as a monolithic complex, without the possibility of free modification of its individual element, which leads to low scalability of such systems;

- each module is able to provide a specific set of methods and capabilities for other modules - the so-called module interface, which makes it possible to standardize inter-module interaction protocols;
- the design interface of the module is an invariable element (and can only be changed when the system is redesigned), but the way of implementing this interface may change. This makes it easy to test new features of the module by gradually changing the way the interface is implemented.

The implementation of the modular system requires the correct distribution of the tasks performed between the modules, as well as the optimality of interaction between them, which is achieved by preliminary analysis and design. The task at its core requires a modular solution, as it must provide functionality at the level of the class library for future use.

The data source encapsulates the logic behind In ASP.NET MVC applications. As it is possible to operate a set of entities and models, for management of which many classes of repositories are created. Unit Of Work pattern was used for working with the database. The Unit Of Work pattern simplifies the work with different repositories and guarantees that all repositories use the same data context.

### *System architecture*

A user interface was developed to implement the database created in Microsoft SQL Server 2014 DBMS, Entity framework object-relational mapping was used to access the data, ASP.NET web framework was used to facilitate the construction of the web application. Apache Solr solution was used as a search engine to perform a full-text resource search.

User interface was built using a server engine for rendering Razor Template.

The server side is designed on the basis of OOP and consists of classes that are combined into packages or namespace. JetBrains - Resharper was used to simplify the development and code generation process.

C # was used as the programming language for the system implementation.

The project was designed using the MVC template architecture. C # classes are divided into packages: controllers, models and views.

The following metadata schema for the resource (Fig. 2) was defined when developing the system: unique identifier; title (up to 254 characters); a concise description of the content (up to 1000 characters); the language of the resource; the level of secrecy; format (file format in which the dataset is available); the ID of the directory to which the resource belongs; status of the resource; physical path to a resource on disk.

Developing of metadata resource schema needs security mechanism for access to directories and specific resources.

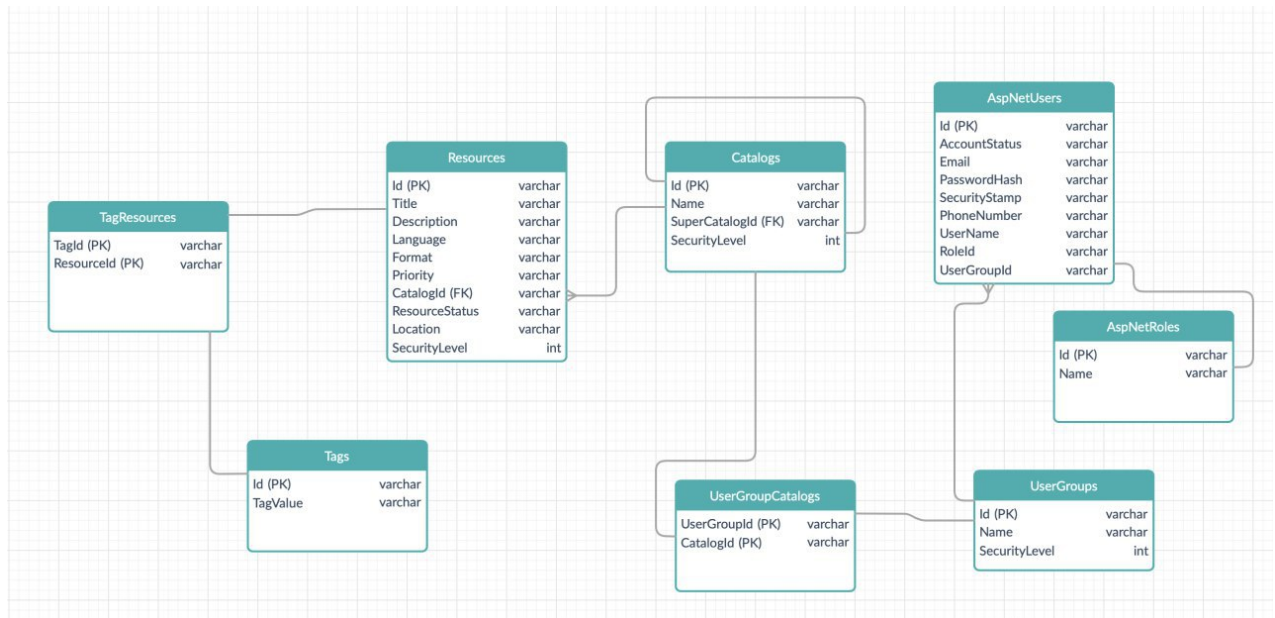


Fig. 2 – Database structure

To ensure the security of resources, it was decided to specify a list of user groups that will have access to the directory when creating the latter. However, another mechanism has been developed to adjust the level of access for individual resources. To do this, when creating a resource, the privacy level is specified. The privacy level is a resource attribute, or a number from one to ten. The same mechanism is present when creating custom groups. Each user inherits the privacy level of the group to which he or she belongs. If the user's privacy level is greater than or equal to that of the property, then the user will have access to the resource (even though the directory to which the property belongs does not allow access to that group of users).

Also, in order to improve quality of the system and facilitate project support in the future, the code style was adhered the SOLID principles.

### *System description*

An entire layer of repository classes was enclosed to interact with the database this encapsulates the logic behind query execution.

Repositories are classes that implement the Dao pattern. Classes also serve to implement business logic.

Views are special classes that are responsible for displaying customer information. .NET representation of this system is built using Razor Template technology, where web page rendering is server-side.

The central entity in the database is the Resources table (see Fig. 2). This table is based on the STI (Single Table Inheritance) pattern. This means that one table serves to aggregate data from the types in the inheritance hierarchy. This table contains the metadata of all system resources. The Users table contains all the user settings. The Catalogs table is used to store directory information. The UserGroups

table contains user groups to differentiate access to resources. The UserGroupCatalogs table is a table of links between user groups and resource directories. The Tags table contains keywords that you can use to search for resources. The TagResources table is a table that links a table of tags and resources. TheAspNetRoles table contains information about the role of the user in the system. AspNetRoles and AspNetUsers tables are system tables that use ASP.NET framework in their work.

In general, the system accomplishes the following tasks: registration of the user in the registry; user authentication and authorization; creation of electronic resource in the registry; registration of the resource in the registry; storing metadata resources in the registry; linking resource directories and user groups, uploading a resource to a local file system; viewing the resource by direct link; Uploading a resource to a local computer possibility of full text search on the content of documents.

### *Conclusions*

In this work was built a scheme of metadata registers and developed web-system for managing the registry of electronic information resources. Web interface for access to the register of EIR of the department has been developed too. A mechanism for regulating the level of secrecy of resources was realized. The web-system are provided efficient expansion of the functionality of the already installed architecture for the system updates.

### *References*

31. Eremenko T. V. Cataloging of Internet Resources: Problems and Solutions (Experience of the Web Resources Catalog's Creation in the Samara Regional Universal Research Library). - 2002. - № 5. - p. 53-68.
32. About the National Register of Electronic Information Resources: Decree of the Prime Minister of Ukraine # 326 of March 17, 2004.
33. Danilin A.V. Standards and unified information technology architecture. - Microsoft Press, 2003.  
<http://www.microsoft.com> /Ukraine/Government/Analytics/Integration Technologies /Standards.aspx.